AN ENHANCED SUPPORT VECTOR REGRESSION-AFRICAN BUFFALO OPTIMISATION ALGORITHM FOR ELECTRICITY TIME SERIES FORECASTING

INUSA SANI MAIJAMA'A

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2023

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

INUSA SANI MAIJAMA'A

calon untuk Ijazah **PhD**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"AN ENHANCED SUPPORT VECTOR REGRESSION -AFRICAN BUFFALO OPTIMISATION ALGORITHM FOR ELECTRICITY TIME SERIES FORECASTING"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : **01 November 2023.**
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
01 November 2023.

Pengerusi Viva: *(Chairman for VIVA)*	**Assoc. Prof. Dr. Mazni Omar**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Prof. Dr. Rozaida Ghazali**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Assoc. Prof. Dr. Siti Sakira Kamaruddin**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Assoc. Prof. Dr. Yuhanis Yusof**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Dr. Mohamad Farhan Mohamad Mohsin**	Tandatangan *(Signature)*

Tarikh:
(Date) ***01 November 2023***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis. Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

Abstrak

Keupayaan meramalkan masa depan membolehkan keputusan termaklum dan perancangan strategik dilakukan. Dalam ekonomi moden, keseimbangan pengeluaran dan penggunaan elektrik amat penting. Ramalan beban melibatkan anggaran penggunaan elektrik pada masa hadapan, dan ia dipengaruhi oleh kepadatan populasi, cuaca, polisi dan aktiviti sosio-ekonomi. Regresi Vektor Sokongan (SVR) digunakan secara meluas dalam ramalan tetapi keberkesannnya bergantung kepada nilai penyesuaian, toleransi, parameter kernel. Kajian ini mencadangkan algoritma hibrid yang dinamakan sebagai SVR- Algoritma Optimisasi African Buffalo (ABO). Proses pengoptimuman ABO melibatkan empat fasa; SVR-ABO, SVR-*pop*ABO, SVR-*explr*ABO, dan SVR-*explt*ABO. SVR-ABO menggunakan ABO untuk mengoptimumkan hiperparameter SVR. Manakala, SVR-*pop*ABO meningkatkan kepelbagaian ABO menggunakan fungsi huru-hara, dan *explr*ABO menggunakan penerbangan levi untuk mencari dan mengatasi optima tempatan yang lebih baik. Di samping itu, *explt*ABO menghalang penumpuan pramatang. Empat hibrid ini mewakili penambahbaikan progresif ABO klasik untuk mengoptimumkan hiperparameter SVR. Menggabungkan algoritma yang dipertingkatkan menghasilkan SVR-*e*ABO, yang kebolehan ramalannya telah dinilai menggunakan MAE, MAPE, RMSE, PA dan R^2. Dinilai menggunakan set data penanda aras, SVR-*e*ABO mencapai ketepatan tinggi, mengatasi SVR standard dan varian SVR berasaskan pengoptimuman lain seperti SVR-PSO, SVR-ABC, SVR-CS, dan SVR-GA. Sebagai contoh, SVR-*e*ABO mencapai ketepatan 98.51% pada set data Household, 98.15% pada set data Turkey, 91.17% pada set data Appliances, dan 96.52% pada set data Panama. Algoritma SVR-*e*ABO yang dicadangkan mempunyai implikasi yang signifikan untuk meningkatkan ketepatan ramalan beban, membolehkan pengurusan grid elektrik yang lebih cekap, dan memudahkan pembuatan keputusan yang berinformasi untuk penyedia dan pengguna tenaga

Kata Kunci: Regresi vektor sokongan, Pengoptimuman kerbau afrika, Penerbangan levi, Algoritma mcCulloch, Ramalan siri masa

Abstract

Time series forecasting enables informed decision-making and stakeholder benefit. Electricity production-consumption balance is vital in modern economies. Load forecasting predicts electricity consumption, influenced by factors like population, weather, policies, and socio-economic activities. Support Vector Regression (SVR) is a widely used regression technique, but its efficacy depends on optimal tuning of parameters, which is challenging. This study proposes a hybrid approach combining SVR and the African Buffalo Optimization (ABO) algorithm. The classical ABO algorithm faces limitations in population initialization, exploration, and exploitation. Therefore, enhancements have been made to these stages to improve performance. The study presents a series of hybrid algorithms that leverage ABO to optimize SVR hyperparameters. SVR-ABO uses the classical ABO approach. SVR-*pop*ABO enhances population diversity using a chaotic function. SVR-*explr*ABO includes Lévy flight to improve exploration and overcome local optima. SVR-*explt*ABO modifies the exploitation mechanism to prevent premature convergence. These four hybrids represent a progressive refinement of the classical ABO for optimizing SVR hyperparameters. Combining the enhanced algorithms results in SVR-*e*ABO, whose forecasting ability has been assessed using MAE, MAPE, RMSE, PA and R^2. Evaluated using benchmark datasets, SVR-eABO achieves high accuracy, surpassing standard SVR and other optimization-based SVR variants like SVR-PSO, SVR-ABC, SVR-CS, and SVR-GA. For instance, SVR-eABO achieved 98.51% accuracy on the Household dataset, 98.15% accuracy on the Turkey dataset, 91.17% accuracy on the Appliances dataset, and 96.52% accuracy on the Panama dataset. The proposed SVR-eABO algorithm holds significant implications for improving load forecasting accuracy, enabling more efficient electricity grid management, and facilitating informed decision-making for energy providers and consumers.

Keywords: Support vector regression, African buffalo optimisation, Time series forecasting

Acknowledgement

I am immensely grateful to Allah S.W.T for His blessings, including good health, strength, and resilience throughout this transformative journey.

My deepest appreciation goes to my primary supervisor, Prof. Madya Dr. Yuhanis binti Yusof, whose unwavering support and invaluable guidance have shaped my research. Dr. Yuhanis has not only provided diverse methodologies for tackling research problems but has also instilled in me virtues such as courage, perseverance, and patience. I am profoundly grateful for her understanding and exceptional supervision.

I also express my deep gratitude to my co-supervisor, Dr. Mohamad Farhan Bin Mohamad Mohsin, whose patient oversight and enthusiastic encouragement have significantly contributed to the progression of my research.

I am fortunate to have worked under the exceptional advice and guidance of Prof. Dr. Ku Ruhana Ku Muhammad. Her kindness and dedication in fostering a supportive environment have promoted collaboration, innovation, and personal growth. I express my deepest appreciation for Prof. Dr. Ku Ruhana's remarkable leadership, unwavering support, and commitment to creating a positive and enriching learning experience for all in the lab.

I am immensely grateful to TETFUND for their generous sponsorship of my PhD studies. Their financial support has been instrumental in enabling me to pursue my research goals and participate in valuable academic opportunities. I extend my heartfelt appreciation to TETFUND for their unwavering commitment to supporting higher education and research.

Lastly, I offer my sincere regards and blessings to my family and friends who supported me in various capacities. I remain humble and grateful for their unwavering support in my life. I am deeply thankful to Allah for guiding me through this scholarly pursuit and for the opportunities that lie ahead.

To the most Professional,
Most Patient, and
Most Caring Supervisor

Prof. Dr. Yuhanis bint Yusof

Simply, You're the Best.

Table of Content

List of Tables

List of Figures

List of Algorithms

CHAPTER ONE

INTRODUCTION

This chapter serves as a foundational overview of the study, establishing the context and significance of the research on electric load forecasting using machine learning algorithms. It delineates the critical aspects of multivariate time series forecasting and highlights the challenges associated with accurately forecasting electricity consumption. The chapter further elaborates on the role of Support Vector Regression and the African Buffalo Optimization algorithm in addressing these challenges. By outlining the background, research questions, objectives, and limitations of the study, this chapter aims to provide a comprehensive framework for understanding the subsequent analysis and findings presented throughout the thesis.

1.0 Background Study

Accurate forecasting of electric load holds significant importance in facilitating decision-making processes pertaining to power unit commitment, economic load dispatch, power system operation and security, contingency scheduling, among others. Previous studies have highlighted that even a 1% increase in forecasting errors for electric load can result in an additional operational cost of £10 million (Dong et al., 2018). Conversely, reducing forecasting errors by 1% can yield notable operational benefits. Consequently, there is a strong impetus to explore more accurate forecasting models and novel intelligent algorithms to achieve satisfactory load forecasting outcomes. This pursuit aims to optimize the decisions pertaining to electricity supplies and load plans, enhance the efficiency of power system operations, and ultimately mitigate system risks within a manageable range. However, the complexity of electric load forecasting arises from various factors, including energy policy, urban population dynamics, socio-economic activities, weather conditions, holidays, and other pertinent variables. The presence of seasonality, non-linearity, and chaotic patterns in electric load data further complicates the task of load forecasting.

Numerous electric load forecasting models have been proposed with the aim of continuously improving forecasting accuracy. These models can generally be categorized into two types: those based on statistical methodologies and those utilizing artificial intelligence (AI) technology. Statistical models, such as ARIMA models, regression models, exponential smoothing models, Kalman filtering models, and Bayesian estimation models, rely on historical data to identify linear relationships among different time periods. However, these statistical models are inherently limited by their theoretical assumptions and are only capable of effectively handling linear relationships between electric loads and the aforementioned factors. Consequently, their forecasting performances often fall short of satisfactory results.

Artificial intelligence (AI) technologies, including Artificial Neural Networks (ANNs), Expert System models, and Fuzzy Inference systems, have gained significant popularity in improving the performance of electric load forecasting owing to their superior ability to handle nonlinear processing. However, it is important to note that AI models, including hybrid and combined models, also possess their own limitations. These limitations include computational time requirements, challenges in determining structural parameters, and the potential for getting trapped in local minima.

The Support Vector Machine (SVM) is a powerful algorithm that has found widespread application in various scientific fields, including machine learning (Che et al., 2017; Ouahilal et al., 2017). It was introduced by Vapnik, Boser, and Guyon in 1992 (Boser et al., 1992), and further developed by Corinna and Vladimir in 1995 (Corinna & Vladimir, 1995). The success of SVM can be attributed to its strong theoretical foundation. The algorithm's ability to generalize stems from its effectiveness in solving classification problems involving non-linearly separable data in high-dimensional spaces. This capability has made SVM one of the most suitable algorithms for a range of data mining tasks, including classification and regression.

In forecasting analytics, SVM is commonly used for classification or regression tasks, depending on the nature of the desired output. When the output is categorical, the technique is referred to as SVM, whereas when the output consists of continuous numerical values, it is typically referred to as Support Vector Regression (SVR) (Moon et al., 2018; Ouahilal et al., 2017).

The support vector regression (SVR) model, known for its remarkable nonlinear processing capabilities and utilization of high-dimensional mapping and kernel computing techniques, has shown remarkable application results on various regression tasks. The empirical evidence highlights that an SVR model, when equipped with accurately computed parameters through the use of swarm-based algorithms, can deliver highly satisfactory forecasting performances.

Despite the success of SVR, its effectiveness in a given task is highly dependent on the values of its hyperparameters, namely C, γ, and ε (Bing et al., 2018; Jiang et al., 2018; Sarhani & El Afia, 2015). Therefore, studies focusing on the optimization of these hyperparameters are crucial to ensure optimal performance of the algorithm.

In this respect, this study proposes to hybridize SVR algorithm with a population-based optimisation algorithm named African Buffalo Optimisation (ABO), which was introduced by Odili, Kahar and Anwar (Odili et al., 2015). The ABO algorithm is based on the searching and foraging behaviour of African buffalo. This algorithm draws its inspiration from observing a specie of African wild cows called African Buffalos in their quest for grazing pastures in the African forests. The animal is in competition with other herbivorous animals which most times require less intake of pastures than this large animal with big appetite. A lot of ingenuity is required if she is to survive the competition and sometimes the hostility of African lions and human hunters. The ABO algorithm models the animal's ingenuity in navigating her way through several thousands of kilometres in the vast African forests with the sole aim of tracking the wet seasons in different locations where it could satisfy its appetite. Tracking the best position and speed of each buffalo ensures adequate exploitation of the search space and tapping into the experience of other buffalos as well as that of the best buffalo enables the ABO to achieve adequate exploration. The algorithm has been rigorously tested by the authors based on Symmetric Traveling Salesman's Problem and proven to be effective in comparison to other well-established swarm-based optimisation algorithms like Particle Swarm Optimisation (PSO) (Eberhart & Kennedy, 2016; Jia, 2015; Yan et al., 2012), Ant Colony Optimisation (ACO) (Deng et al., 2014; Gündüz et al., 2015), Honey Bee Mating Optimisation (HBMO) (Marinakis et al., 2011) and HPSACO (Odili, Kahar, Noraziah, et al., 2017).

However, standard ABO algorithm operates by having its learning parameters (lp_1 & lp_2) set prior to execution (Odili et al., 2015). As these two learning parameters controls both personal and global best of the buffaloes, this could easily lead the algorithm to have poor exploration and exploitation. Similarly, the position update mechanism is solely controlled by a preset lambda parameter which obviously does not have element of diversity as it is also arbitrarily set prior to execution, this could easily lead the buffaloes to be trapped in local optima (El-Ashmawi, 2018). To address the mentioned limitations of ABO algorithm, consequently, to increase the ability of SVR in its generalisation, there is a need to enhance the ABO algorithm. This can be achieved by mitigating the problem of the learning parameters and, by making the exploration controlling parameter to be from an efficient random source for effective exploration and exploitation.

This study evaluates the proposed hybrid algorithm in the power sector due to the trend of deregulation especially as witnessed in developing countries (Hall & Nguyen, 2017; Weron, 2014). Deregulation in electricity market chain has already become the mainstream approach in the developed world. Monopoly and absolute control of the sector by government bodies is becoming more obsolete by day throughout the world. The electricity sector has witnessed total overhauling in terms of becoming standard market that has both vertical and horizontal integration of all related sectors from generation, distribution to consumption.

1.1 Time Series Forecasting

Time series forecasting is one of many interesting areas in various fields. It offers the ability to forecast future which can be relied upon for making informed decision or planning an action to be taken for the benefit of stakeholders. Time series forecasting relies on historical data as the main input in order to be able to forecast the future. The importance of time series forecasting has been witnessed in various domains including but not limited to electricity consumption (Dung et al., 2021), daily natural gas consumption (Wei et al., 2019), Air passenger flow (Ashraf et al., 2021), heat load (Bergsteinsson et al., 2023), wind power (Ashraf et al., 2021), solar energy (Cabello-López et al., 2023), seasonal stream flow (Petry et al., 2023), and oil price (Ellwanger & Snudden, 2023).

1.2 Multivariate Time Series Forecasting

The field of time series analysis and forecasting research has been active for a long time especially in fields of statistics, signal processing, econometrics, and mathematical finance, and there have been several articles published in this field (Agrawal et al., 2018; Fu et al., 2015; Huo et al., 2017; Kong et al., 2018; Lang et al., 2018). However, researchers were only just concern scalar time series in most of the papers. In principle, according to the Takens' embedding theorem, scalar time series are generally sufficient to reconstruct the dynamic of the underlying systems if there are enough delayed coordinates to be used. But in practice, this may be incorrect (Lang et al., 2018). Consequently, in practical problems it cannot be sure whether any given scalar time series are sufficient to reconstruct the dynamics. Furthermore, it is anticipated that there may be some substantial advantages if several different time series are used, especially when the system is noisy. Multivariate time series data are common in practice: physiological data, electroencephalograph (EEG) data, economic data, electric load forecasting data and so on.

1.3 Electric Load Forecasting

Electric load forecasting (ELF) is usually considered based on three forecasting horizons namely short term (Al-Musaylh et al., 2018; Avatefipour & Nafisian, 2018; Bandyopadhyay et al., 2018; Yusof & Mustaffa, 2015), medium-term (Bouktif et al., 2018) and long term (Agrawal et al., 2018) as proposed by Mocanu, Nguyen, Gibescu and Kling (Mocanu, Nguyen, Gibescu, & Kling, 2016). Though there is no clear distinctive boundary for each of the three categories, yet some approximate threshold values are being used.

Short-term load forecasting (STLF) generally refers to the type of forecasting that spans over a short period of time. The period can be from few minutes to few days ahead of present time (Fan et al., 2021; Masum et al., 2018). This type of forecasting is usually used by electricity producing firms for day-ahead and intra-day trading, and for day-to-day market operations. Hence, this study adopts the forecast electricity load.

Medium-term forecasting (MTLF) refers to the type of forecasting that spans from few days to few months ahead (P. Su et al., 2017). This type of forecasting is more useful to stakeholders for operations that is of less frequent in nature like risk management, derivatives pricing and balance sheet calculation. Also, this type of forecasting is usually more focused on how the prices are distributed rather than single point pricing forecast .

Long-term load forecasting (LTLF) refers to type of forecasting period that spans from few months to several years (Sarhani et al., 2018). This type of forecasting is usually used for long-term investment profitability analysis like making decision to construct a new power plant or not.

Various techniques are employed for ELF. These techniques can be broadly categorised into statistical and computational intelligence as stated by (Dong et al., 2018). The most popular among statistical techniques are time series techniques that comprises of Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA) and its variants (Mat Daut et al., 2017) and Exponential Smoothing models (Hermias et al., 2018). However, statistical techniques were found to be poor of performance (Conejo et al., 2005; Ugurlu et al., 2018) due to their inability to deal with complex and non-linearity in multivariate data (Dong et al., 2018; Hamzaçebi, 2008; Li et al., 2018; Mustaffa et al., 2015). Hence, they are not suitable to be used for electric load type of data which is complex and non-linear (Mocanu, Nguyen, Gibescu, & Kling, 2016; Salkuti, 2018). These highlighted shortcomings of statistical techniques prompt researchers to explore computational intelligence methods. Among the two popular computational intelligence methods in dealing with time series forecasting problems are the Artificial Neural Network (ANN) and Support Vector Machines (SVM) (Guo et al., 2015; B. Huang et al., 2018; Mocanu, Nguyen, Gibescu, Larsen, et al., 2016). ANN forecasting ability relies on estimation of values for an unknown function which most times yield good forecasting result (Jaddi et al., 2017). However, despite positive forecasting outcomes from Artificial Neural Network (ANN), the method suffers from time-consuming training and vulnerability to overfitting (Eapen & Simon, 2018).

In order to overcome the problems of ANN, Support vector Machines (SVM) introduced by Vapnik in (Corinna & Vladimir, 1995) were proven to be most effective due to its adoption of Structural Risk Minimisation (SRM) approach (Al-Musaylh et al., 2018). This approach focuses on minimising the generalisation error instead of minimising training errors as done by ERM. This makes SVM able to overcome the problem of over-fitting therefore capable to achieve good generalisation.

The Support Vector Regression (SVR), as a of SVM, that is meant for regression task (Al-Musaylh et al., 2018; Chuang et al., 2002), has proven to be powerful in the field of load forecasting (Azad et al., 2018; Caraka et al., 2018; Dong et al., 2018; Jungwon et al., 2018; Li et al., 2018; Li et al., 2018; Moon et al., 2018a; Sarhani et al., 2018; Su & Chawalit, 2018; Sun et al., 2018; Velasco et al., 2018; Yang et al., 2019; S. Zhang et al., 2019; Zhang, 2018). However, the generalisation performance of SVR relies on two parameters values (Peng et al., 2016; Sarhani et al., 2018) which are cost error (C), tube size (ε) and gamma (γ), in the case whereby Radial Basis Function (RBF) kernel is selected as an additional parameter (Hu et al., 2014; Humeau et al., 2013; Iliya et al., 2015; Peng et al., 2016; Sarhani et al., 2018; Sarhani & El Afia, 2015). Manual selection of these parameter value can be a complex task. This necessitates the need to find best approach in determining the optimal value for parameters of SVR in order to get the optimal generalisation that will eventually lead to better accuracy of the algorithm.

Three major approaches became popular in optimisation process viz: cross-validation (CV), grid search and metaheuristics techniques. As for the cross-validation and grid search techniques, it has been reported that they are computationally expensive and usually reported high error rate (Bing et al., 2018; Che et al., 2017; Yusof & Mustaffa, 2016). This make CV and grid search techniques to be a bad choice for parameter value optimisation for SVR, hence led to adoption of metaheuristics techniques.

The use of metaheuristic techniques is being widely reported in literature as a means of determining optimal values for SVR parameters through hybridisation (Dong et al., 2018). The hybridisation proves to be yielding positive results in terms of obtaining optimal values for SVR parameters hence produce better generalisation as can be seen in (Chou et al., 2017; Chou & Pham, 2017; Chou & Truong, 2019; Li et al., 2018; Li et al., 2018; Sermpinis, Stasinakis, & Hassanniakalager, 2017). Among of the effective optimisers includes GA (Xie et al., 2017) and PSO (Mohanad et al., 2018) However, recently a new optimisation algorithm namely African Buffalo Optimisation (ABO) has also shown a promising result.

This study therefore investigated the effectiveness of ABO in optimising SVR in multivariate forecasting.

1.4 Problem Statement

In spite of the aforementioned merits attributed to the Support Vector Regression (SVR) algorithm, as discussed in the background section, it is confronted with a significant obstacle pertaining to hyperparameter optimisation.

SVR necessitates the careful tuning of three pivotal hyperparameters: Punishment factor (*C*), Tube size (ε) and the kernel parameter (γ) (Iliya et al., 2015; Peng et al., 2016; Sarhani et al., 2018; Sarhani & El Afia, 2015).

The arduous nature of pinpointing the optimal parameter values in SVR underscores the compelling need to explore novel methodologies that can surmount these challenges and deliver enhanced performance. Hence, SVR has been combined with different metaheuristic algorithms like Genetic Algorithm (GA) (Xie et al., 2017), Particle Swarm Optimisation (PSO) (Jalalifar et al., 2019; Yao & Mao, 2023) and Grasshopper Optimisation Algorithm (GOA) (Barman et al., 2018). However, Genetic Algorithm relies on the initialisation of various parameters like population, fitness function, mutation rate, cross-over rate and selection method (Avatefipour & Nafisian, 2018; Wei et al., 2018).

Similarly, PSO performance is sensitive to its parameters, such as the inertia weight, cognitive and social learning factors, and population size. Selecting appropriate parameter values can be challenging and may require some trial and error. In addition, PSO has the potential for premature, and an inherent problem of slow convergence (Avatefipour & Nafisian, 2018; Lai & Zhou, 2019).

Likewise, GOA encounters a challenge of becoming entrapped into local optima as iterations progress. This is attributed to a reduction in the diversity of the swarm. Additionally, GOA lacks a mechanism to preserve the elite grasshoppers discovered thus far within each index, resulting in a compromised exploitation ability and diminished convergence rate for the algorithm (Ingle & Jatoth, 2023).

The issues delineated concerning Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Grasshopper Optimisation algorithm (GOA) techniques have the potential to adversely influence the efficacy of Support Vector Regression (SVR) models generated through either of these optimisation methods. Consequently, the forecasting accuracy of any SVR model constructed using these optimisation approaches might be compromised (Bing et al., 2018).

African Buffalo Optimisation (ABO) which has gained significant popularity across diverse optimization domains. Examples of its applications include team formation (El-Ashmawi, 2018), the Traveling Salesman Problem (TSP) (Odili, Kahar, Noraziah, et al., 2017), biodiversity conservation area selection with constraints (Almonacid et al., 2017), and PID controller parameter tuning (Odili, Kahar, & Noraziah, 2017). The ABO algorithm has the advantage demonstrated rapid convergence, and effective tracking of the best position over other similar algorithms like PSO, GA, and Cuckoo algorithms as put forward by its authors (Odili & Kahar, 2015). However, these attributes have yet to be empirically validated when employing the ABO algorithm as an optimizer for Support Vector Regression (SVR) models. Despite its utilisation in various domains in literature, the ABO algorithm encounters difficulties pertaining to population initialisation, exploration, and exploitation (Arif et al., 2022; J. B. Odili, Kahar, Noraziah, et al., 2017b; Peace Igiri et al., 2018; S. Zhang et al., 2019; Zhu et al., 2020).

ABO population initialisation mechanism uses simple random distribution. However, using simple random distribution as a method of population initialisation leads to less diversity of population sample in search space, hence hampering with the convergence speed and prevents population to escape local optima entrapment (Arif et al., 2022; Zhang et al., 2019).

Despite several attempts made by researchers to improve population initialisation segment of ABO algorithm as can be found in (Algaphari, 2023; Barnwal et al., 2023; Jiang, Tianhua-Zhu & Deng, 2020; Mishra, 2022). However, none has tried employing Tent map function for population initialisation in ABO algorithm.

As for the exploration process, researchers employed chaotic map with logistic map function to modify exploration process of ABO algorithm (Igiri, Singh, & Bhargava, 2019b). However, logistic map function relies on Chebyshev-type distribution that require the function to go through multiple search iterations which are unnecessary as argued by Lu (Lu et al., 2014). This makes the exploration process to repeat similar visited position in the searching space and most of times lead to premature convergence.(Lu et al., 2014). There are evidently several efforts made by researchers to improve the exploration function of ABO as can be found in literature (Algaphari, 2023; Igiri, Singh, & Poonia, 2019; Jiang, Tianhua-Zhu & Deng, 2020; Sheeba et al., 2023), yet none has tried using McCulloch based levy flight function for the enhancement of ABO exploration mechanism.

On top of that, the exploitation process in ABO algorithm uses manual assignment of exploitation control values, which are mostly obtained through a more of trial-and-error approach. This arbitrary approach of manual assignment of exploitation process values could result into missing the ideal values needed for an optimal exploitation process, that eventually lead to local optima entrapment (Igiri et al., 2019b; Odili et al., 2017). Like in the case of exploitation mechanism of ABO, several attempts have been made as can be found in literature (Igiri, Singh, & Poonia, 2019; Jiang, Tianhua-Zhu & Deng, 2020; Mishra, 2022). However, none among the mentioned researchers employs the use of Tent-map function for enhancement of exploitation mechanism in ABO algorithm.

Conclusively, this study proposes to enhance the ABO prior to its deployment as an optimiser for SVR algorithm. Prior to that, the SVR will be optimised using the classical ABO.

1.5 Research Questions

The questions to be answered by this research are as follows:

1. How to automatically optimise SVR algorithm's hyper-parameter using ABO algorithm?
2. How to design ABO with diverse solution population using Tent-map based chaotic function to increase convergence speed?
3. How to enhance the exploration ability of ABO algorithm using Mc-Culloch based Levy flight function to avoid premature convergence?
4. How to automatically tune the ABO's exploitation process parameters using Tent-map based chaotic function to avoid being entrapped in local optima?
5. How to evaluate the proposed SVR-ABO and other enhancements made on ABO algorithm?

1.6 Research Objectives

The aim of this research is to propose a multivariate time series forecasting algorithm based on the integration of SVR and ABO algorithm. The following specific research objectives are to be fulfilled:

i) To design an optimised SVR algorithm using classical ABO algorithm.

ii) To design ABO population initialisation function using Tent-map based chaotic function for maximum population diversity in the search area to increase convergence speed.

iii) To reformulate ABO exploration function using Mc-Culloch based Levy flight function to enable the algorithm to be more resilient to premature convergence by optimal placement of buffaloes in a wider search space.

iv) To reformulate ABO exploitation function using Tent-map based chaotic function to prevent falling into local optima.

v) To evaluate the proposed enhanced SVR-ABO algorithm against existing optimised SVR algorithms.

1.7 Scope and Limitation of the Study

The scope of this study is based on multivariate short-term of electric load forecasting that can be determined based on consumption. The study intends to use four (4) datasets as follows:

i. Individual Household Electricity Consumption dataset (Hebrail, Georges and Berard, 2012; Sinha et al., 2021).
ii. Appliances Energy Forecasting dataset (L. Candanedo, 2017).
iii. Turkey electricity consumption (Tutun, 2016).
iv. Panama Electric load consumption (Madrid & Antonio, 2021).

This study aims to hybridise a machine learning technique (i.e., SVR) with Swarm Intelligence (SI) method for forecasting task with attention on SVR parameter tuning with SI method. However, the study is limited to employing ABO algorithm to optimise SVR algorithm's parameters towards building an enhanced algorithm for an electric load forecasting purpose.

1.8 Significance of Study

The primary contribution of this research lies in its advancement of the existing body of knowledge, specifically in the context of hybridizing SVR-ABO and enhancing the ABO method. Notably, the researcher has developed an enhanced ABO algorithm that encompasses three crucial stages: population initialization, exploration, and exploitation.

The novelty of this approach is underscored by its potential to make accurate forecasts regarding future electricity consumption. By employing an optimized multivariate time series forecasting algorithm, SVR-ABO, the algorithm effectively learns the consumption patterns of consumers. This has significant implications for efficient management and effective future planning in the power generation industry. By reducing resource losses and minimizing wastage production, the algorithm can enhance profitability and minimize excess production, which is particularly crucial given the inherent limitations of storing excess electricity.

The proposed algorithm draws upon the generalization ability of SVR and leverages the rapid convergence speed of the ABO algorithm. This unique combination yields an optimal electric load forecasting algorithm that is both diverse and efficient in its exploration and exploitation processes. By integrating these elements, the algorithm offers a novel and promising approach to electric load forecasting, contributing to the advancement of knowledge in this field.

1.9 Summary

In this chapter, the introduction of the research domain is provided as part of the background of the study. The introduction of the key algorithm, upon which this study has been built, is also presented, along with the introduction of the optimisation algorithm. The strengths and shortcomings of the key algorithm, SVR, are highlighted. Furthermore, the need for hybridization with a swarm intelligence-based algorithm is presented. The chosen SI algorithm, ABO, is presented, along with the justification for the hybridization based on its strengths. The weaknesses of ABO algorithm are addressed in the problem statement. This lays the foundation for the formulation of research questions that guide the development of the research objectives in this study. Finally, the scope, significance, and limitations of the study are also presented.

CHAPTER TWO

LITERATURE REVIEW

As highlighted in previous chapter, there are two major approaches employed in forecasting namely Statistical and Machine learning methods. With respect to that matter, this section reviews the existing work that employs different Statistical methods as well as Machine learning methods. Review of SVR, and hyper-parameter optimisation techniques were also provided. Lastly, African Buffalo Optimisation algorithm was discussed, highlighting its strengths & weaknesses, as well as various literature that mentioned its application fields.

2.1 Multivariate Time Series

A time series pertains to a sequential arrangement of values that are observed at consistent intervals throughout a predetermined temporal span. For a model to be considered multivariate, it must involve other related time series factors that affect the target (Ziel, 2015). For example, in a study by Javedani et al., the authors considered temperature as an influencing factor of power consumption in companies located in Johor, Malaysia (Sadaei et al., 2019). In other studies, such as those conducted by Cinar and Madhavi, and Madhavi, temperature, humidity, solar radiation, and traffic flow were considered as influencing variables for electricity load consumption (Cinar et al., 2018) (Madhavi et al., 2017).

Cheung et al., (2018) considered temperature as a factor that makes data to be considered as multivariate. This research employs multivariate time series analysis, considering variables such as temperature and humidity in building the forecasting model.

2.2 Electric Load Forecasting Methods

Accurately forecasting electric load consumption plays a critical role in ensuring efficient energy management, grid stability, and informed decision-making across various sectors. The challenges associated with load forecasting necessitate the utilization of effective methods capable of capturing the intricate dynamics of electricity demand. This sub-section aims to delve into an examination of the methods employed in electric load forecasting, with the objective of providing valuable insights into their applicability, strengths, and limitations.

The selection of appropriate forecasting methods is contingent upon the specific requirements and scenarios encountered within the field. Several pivotal scenarios underscore the importance of precise load forecasting, encompassing optimizing power generation and distribution, managing grid stability, facilitating energy trading and market operations, integrating renewable energy sources, supporting demand response programs, and enhancing smart grid management.

The field of forecasting has numerous techniques documented in the literature, which can be classified into two groups: statistical methods and machine methods. The following sections outline the characteristics, strengths, and weaknesses of each category.

2.2.1 Machine Learning Methods

Machine learning offers a range of techniques for electric load forecasting, including Fuzzy Time Series, K-Nearest Neighbour, Artificial Neural Network (ANN), and Support Vector Regression (SVR). Extensive research has been conducted across various forecasting domains to investigate the efficacy of these methods. Following presents a comprehensive review of relevant literature encompassing studies that have leveraged these approaches for electric load forecasting.

(a) Fuzzy Time series

The initial proposal of fuzzy time series was presented by (Qiang & Brad, 1993). Instead of using numerical values, it was based on linguistic values. Thus, after constructing the fuzzy relationships among the samples, the values need to be transformed into numerical values for output. Model developed base-on fuzzy time series has its accuracy dependant on the proper interval chosen (Deb et al., 2017). Although fuzzy time series has been used in various forecasting models, it has some limitations such as determining fuzzy logic weights, membership functions, and optimal rules (Deb et al., 2017).

Sadaei et al., (2017) proposed a novel approach referred to as the SARFIMA-FTS method, combines elements of SARFIMA (Seasonal Autoregressive Fractionally Integrated Moving Average) and Fuzzy Time Series (FTS) models to increase accuracy of seasonal memory time series (SMTS). The study was founded upon higher-order time series and utilised PSO for tuning SARFIMA-FTS hyper-parameters. The validity of the SARFIMA-FTS method was further established through testing on additional STLF datasets from various domains. The results demonstrated that the SARFIMA-FTS method significantly outperformed benchmark models as determined by the SMAPE metric.

Chang et al., (2019) introduced an innovative approach for electricity consumption forecast in Taiwan. The method integrated the Nth Quartile Discretisation Approach (NQDA) within the Fuzzy time series model. Using electricity consumption data from 1996 to 2017, the study demonstrated the superior accuracy of the proposed methodology. It achieved improved forecasting accuracy by minimizing the Root Mean Square Error (RMSE) metric through the incorporation of the NQDA technique.

Silva et al., (2018) use Advanced Fuzzy Time Series (AFTS) as a forecasting method for short-term electricity consumption based on different time intervals. The performance was measured by the MAPE and Inter Quartile Range (IQR) metrics. The result from the investigation suggested that the proposed method had a statistically significant relevance, as evidenced by a p-value of 0.618.

Luferov et al., (2017) developed a method based on fuzzy time series for forecasting electric power load consumption. They analysed the impact of weather on power consumption using data from Smolensk, Russia, spanning from 2016 to 2017. The research revealed a significant correlation between temperature and power usage, highlighting the temperature's influential role. The proposed approach showed improved short-term forecasting accuracy, evaluated using the MAPE metric.

Sadaei et al., (2019) developed a method that combines fuzzy time series with Convolutional Neural Network (CNN) for short-term forecasting of electric power consumption. They used a multivariate dataset of hourly electric load consumption and temperature data to assess the impact of temperature. The CNN extracted features, while fuzzy time series performed the forecasting. Evaluation metrics such as MAPE, RMSE, and APE were used to measure accuracy. The approach aimed to improve forecasting accuracy and understand the temperature-power consumption relationship.

Chen (2016) proposed a hybrid method that hybridised fuzzy time series with LS-SVM and the Global Harmony Search algorithm to build an electricity forecasting model (Chen et al., 2016). The Global Harmony Search algorithm was employed for its search performance efficiency. This method was applied to electric power consumption data from the Guandong province in China. Empirical evidence shows that, the proposed model was able to achieve higher forecasting accuracy and faster convergence. The summary of forecasting methods in the electric load forecasting domain, utilizing the Fuzzy time series approach with various data frequencies, is provided in Table 2.2.

Table 2. 1

Fuzzy Time Series based Approach for Electric Load Forecasting

Authors	**Approach**	**Data Frequency**	**Evaluation Metric**
Sadaei et al., (2017)	SARIMA-FTS	Half-hourly	SMAPE
Chang et al., (2019)	FTS and Nth Quartile (NQDA)	Daily	RMSE
Silva et al., (2018)	AFTS	Weekly, Daily and Hourly	IQR and MAPE
Luferov et al., (2017)	FTS	Hourly	MAPE
Sadaei et al., (2019)	FTS-ANN	Hourly	MAPE, RMSE, APE
Y. H. Chen et al., (2016)	FTS-LSSVM	Monthly	MAPE, MAE and RMSE

(b) K-Nearest Neighbour

The K-Nearest Neighbour (KNN) algorithm is a well-known technique utilised in classification tasks. It functions by evaluating the similarity between different samples within a designated group. The similarity has to be calculated by determining the properties of each sample. Samples with higher degree of semblance are grouped together, newer samples are assigned group based-on exhibited property. Although the KNN algorithm has been employed in forecasting across a wide range of domains, it still faces several challenges such as determining the optimal number of neighbours and similarity computation metric. These parameters are often determined through a trial-and-error approach, this can become burdensome, especially when dealing with a complex objective function (Deb et al., 2017). These issues associated with the KNN algorithm make it unsuitable for electric load forecasting models, where accuracy is of paramount importance.

Wahid & Kim explored the use of KNN for forecasting daily residential energy requirements (Wahid & Kim, 2016). The research methodology employed in their study is rooted in the classification properties of the K-nearest neighbours (KNN) algorithm. The KNN classifier utilizes the Euclidean distance as a measurement metric. Similarity of sample is determined through close resemblance of its properties with the classified previous data sample.

Xianlong et al., (2018), investigated the use of K-NN algorithm to build a forecasting model for electrical energy consumption. The authors were able to determine the effect of unbalanced data on forecasting accuracy of a classification model. To mitigate the discovered issue, the authors utilised KNN algorithm based on computed weights. The model was evaluated using a data obtained from electricity consumption of a household.

(Al-Qahtani & Crone, (2013) developed a forecasting model based KNN electricity demand. The proposed model uses historical electricity consumption data obtained from United Kingdom. The model was trained using hourly electricity consumption data collected in 2004 and was used to predict daily electric consumption for 2005. The results, as measured by the MAPE metric, showed a significant accuracy of 1.8133%.

In another study, authors aimed to develop an accurate and reliable method for predicting electric consumption in order to support energy management and planning. The authors presented a forecasting model based on of KNN and considered the consumption levels of individual appliances and total home electricity consumption using historical data obtained from residential buildings in Maryland and California (Lachut et al., 2015).

In another study, authors presented a novel approach for forecasting electric power consumption in Cameroon Tchuidjan et al., (2014). Multiple-Input Multiple-Output (MIMO) framework is the methodology employed to implement a K-Nearest Neighbour (KNN) model using data spanning from 1972 to 2009. The model performance was evaluated using MAPE metric. The model developed was used for long term load forecast of fifteen years in to future.

Table 2.2 presents a summary of literature that explores various methods utilising KNN as a forecasting technique for Electric load forecasting across different forecasting horizons.

Table 2. 2

KNN-based approach for Electric load forecasting

Authors	**Method**	**Type of Consumption**	**Evaluation Metric**
Wahid & Kim, 2016	KNN	Hourly	Statistical metrics
Xianlong et al., 2018	Balanced KNN	Monthly	MAE
Al-Qahtani & Crone, 2013	Multivariate KNN	Hourly	MAPE
Lachut et al., 2015	KNN, ARMA, Bayesian, SVM	Weekly	Accuracy
Tchuidjan et al., 2014	MIMO-based KNN	Years	MAPE

(c) Artificial Neural Network

Artificial Neural Networks are a group of learning algorithms that draw inspiration from the functioning of biological neural networks. They are widely utilized for approximating values of unknown functions. (Jaddi et al., 2017).

Different type of ANN can be found in the literature depending on the type of the network architecture (Zheng et al., 2019). Few examples of ANN are; Feed-Forward Neural Network (FFNN), Radial Basis Function Network (RBFN), Recurrent Neural Network (RNN). However, among the different mentioned ANN-based architecture, RNN proves to be the most suitable and most popular architecture used for time series related tasks like power load forecasting(Kong et al., 2017; Ugurlu et al., 2018).

Recurrent Neural Network is a type of ANN that employs usage of information from previous feed-forward Recurrent Neural Networks (RNNs) (Kumar et al., 2018). RNN-based architecture can be found in various forms of either Gated Recurrent Units (GRU) or Long-Short Term Memory (LSTM). These mentioned RNN-based variations of architectures were proposed to mitigate the problem of exploding and vanishing gradient that is been associated with RNN (Kong et al., 2017).

Despite the positive predictive outcomes demonstrated by RNN-based methodologies in existing literature, which have shown their ability to address complex and nonlinear problems, these techniques have limitations such as time-intensive processes and susceptibility to overfitting (Hamzaçebi, 2008; Mat Daut et al., 2017; Ugurlu et al., 2018). The susceptibility to overfitting in RNN-based techniques arises from their use of the Empirical Risk Minimization (ERM) approach, which focuses on minimizing training errors (Eapen & Simon, 2018; Mat Daut et al., 2017). Moreover, Recurrent Neural Networks (RNNs) encompass numerous control parameters that require optimization. These parameters include determining the optimal number of hidden layers, selecting the suitable activation function for each layer, specifying the number of training epochs, and choosing the activation function for the output layer (Kong et al., 2018; Yusof & Mustaffa, 2015). This complex parameter landscape makes RNN-based models unsuitable for power consumption forecasting tasks.

Artificial Neural Networks (ANNs) are a widely utilized type of model building technique, employed in resolving various time series forecasting issues across various domains. They have the potential to be synergistically combined with various intelligence techniques, such as swarm intelligence and genetic algorithms. This integration can lead to the generation of highly efficient and optimized outcomes (Ray et al., 2019). ANNs are often employed as forecasting models in domains where the relationship between features of data samples is non-linear.

The primary objective of artificial neural network (ANN) models is to ascertain the optimal weights associated with individual features, thereby minimizing the disparity between actual and target values through utilization of back-propagation process (Rumelhart et al., 1986). The utilisation of back-propagation technique significantly reduced training time of a model while improving forecasting accuracy. ANNs have been utilized as forecasting models for all horizons in electricity consumption forecasting. Electricity consumption is frequently evaluated by taking into account additional time series data, such as temperature and humidity. These factors are instrumental in assessing and understanding the patterns of power usage. Incorporating temperature and humidity information allows for more accurate analysis and forecasting of electric power consumption (Chae et al., 2016; Chitsaz et al., 2015; Hussain et al., 2016; Kelo & Dudul, 2012; Rezaeian-Zadeh et al., 2012).

(d) Support Vector Machines

The Support Vector Machine (SVM) framework was originally conceived and introduced to the academic community by Vapnik and his colleagues at AT&T (Corinna & Vladimir, 1995). They presented this innovative concept as a powerful machine learning algorithm that has since gained significant recognition and adoption in various fields. The pioneering work of Vapnik and his team laid the foundation for the widespread application of SVM in solving classification and regression problems.

The SVM is a classifying algorithm designed to identify a hyperplane that can accurately divide a set of training data into separate classes using linear separation. The points of data that are used to determine the optimal distance between the data points and the margin often referred as hyperplane. These points are , are termed as support vectors. In practical situations, it is uncommon to encounter data that can be perfectly separated by a linear boundary. Therefore, when faced with non-linearly separable data, the SVM employs a technique that involves mapping the data points into a higher dimensional space to identify the hyperplane. This is followed by a process called the kernel trick, which maps the features back to the original space. Additionally, the SVM provides the option for non-correctly classified data points through the implementation of a penalty factor, C, which regulates the acceptable amount of misclassification during optimization while penalizing any errors beyond a predetermined limit.

In addition to its successful application in classification, the SVM algorithm has proven to be effective in regression tasks as well. By aiming to establish a function that minimizes the deviation between output measurements and the cumulative error of input values, Support Vector Regression (SVR) emerges as a powerful technique in forecasting modeling.

2.3 Support Vector Regression

Support Vector Regression (SVR) is a variant of the Support Vector Machine (SVM) algorithm in the field of machine learning. SVR is specifically designed for regression tasks, while SVM is primarily used for classification tasks. SVR was developed by Smola and Scholkopf (Smola & Scholkopf, 2004) to address regression problems and has been proven to be highly effective in classification tasks. The algorithm has become widely used in various applications, including stock market and electric load forecasting (Barman & Dev Choudhury, 2018; Qu & Zhang, 2016; Sermpinis, Stasinakis, Rosillo, et al., 2017). One of the advantages of SVR is the use of kernel functions, which enable the algorithm to perform linear and non-linear approximations. Additionally, SVR is renowned for its superior performance, as it only utilises support vectors to determine the model boundary. Also, due to employing convex objective function, the problem of local minima is eliminated. Another key advantage of SVR is its ability to minimise generalisation error due to adoption of structural risk minima which aims to minimise generalisation error instead of just training error. It should be noted that the authors opted to employ SVR instead of LS-SVR due to its limited generalisation ability (Yan et al., 2017).

2.3.1 SVR Kernels

Kernels are the key functions that allows SVR to be able to exhibits its potentials both in regression and classification tasks. $k(x_i, x_j)$ as defined in Eq. (2.1) represents a kernel function that has a value equivalent to the value of inner product of two vectors $x_i x_j$in the feature space $\varphi(x_i)$ and $\varphi(x_j)$.

$$k(x_i, x_j) = \varphi(x_i) * \varphi(x_j) \tag{2.1}$$

$k(x_i, x_j)$ is a function that is used map the input data in original space into higher dimensional space. This enables the determination of linearly optimal separating hyperplane in the higher dimensional feature space rather than non-linear separating plane in the original input space. There exist four (4) kernel functions that can be found in literature that are mostly used with SVR (Wang & Wang, 2019) namely, Polynomial kernel function, Gaussian Radial Basis kernel function Sigmoid function and Radial Basis kernel function:

1) Polynomial kernel function denoted as:

$$K(x_i, x_j) = (x_i, x_j)^d, d = 1,2, \ldots \quad (2.2)$$

where *d* is the degree of the polynomial.

2) Radial basis function (RBF) kernel denoted as:

$$K(x_i, x_j) = e^{-\gamma} \left(\|x_i, x_j\|^2 \right) \quad (2.3)$$

3) Gaussian radial basis (Special case of RBF) kernel function is denoted as:

$$K(x_i, x_j) = exp\left(\frac{-(x_i - x_j)^2}{\sigma^2} \right) \quad (2.4)$$

where $\sigma^2 > 0$ denotes the kernel width.

4) Sigmoid kernel function is denoted as:

$$K(x_i, x_j) = tanh(b(x_i \cdot x_j) + c) \quad (2.5)$$

where *b* represent the slope and *C* represent the bias of the function.

This research uses the radial basis function (RBF) kernel as it is the most widely used kernel with SVR (Chou & Truong, 2019; Li et al., 2018; Li et al., 2018; Velasco et al., 2018; Zhang et al., 2019). This is due to its ability to outperform other kernels in terms of accuracy and faster training speed in training phase (Al-Musaylh et al., 2018). The effect of this is the reduction of computational time in terms of tuning for optimum hyper-parameters.

2.3.2 Techniques for SVR Hyperparameter Optimisation

Optimisation technique primary’s aim is to help the SVR algorithm to avoid under-fitting or over-fitting during training, which consequently affect the algorithm’s generalisation ability. There are two prominent methods for SVR optimisation task that can be found in literature. These are cross-validation and using swarm intelligence.

(a) Cross-Validation Approach

The most commonly methods found in literature that are used for SVR hyper-parameter optimisation in SVR are Cross-validation and grid search method (Bing et al., 2018). However, cross-validation optimisation methods are computationally expensive and easily falls into local optimum(Mustaffa et al., 2018), hence researchers opt for better approach of using meta-heuristics approaches.

(b) Swarm Intelligence Approach

Due to in adequacy and problems associated with grid and cross-validation approaches of optimisation, meta-heuristics approaches were tried in academia, with the associated promising result, the meta-heuristic approaches are becoming dominant methods for optimisation process in various fields of research. Following are literature where SVR algorithm has been optimised using metaheuristic techniques.

2.3.3 Reviewed literature on Support Vector Regression with Swarm algorithms

Zhang et al., (2019) utilised the Artificial Bee Colony (ABC) algorithm to optimise Support Vector Regression (SVR) parameters for electricity consumption forecasting in China. They emphasised the importance of population initialization in evolutionary algorithms and introduced a tent chaotic strategy and tournament selection procedure to initialize the ABC population and assign values to individual bees. Their proposed approach, ABC-SVR, achieved notable accuracy in terms of Mean Absolute Percentage Error (MAPE) compared to other state-of-the-art, and classical SVR with default parameters.

In a distinct scholarly investigation, the authors have harnessed the potential of the cuckoo search algorithm to effectively optimize Support Vector Regression (SVR) for the purpose of short-term electric load forecasting in residential electricity consumption. However, with the shortcomings of Cuckoo Search algorithm (CSA) of premature convergence and slow convergence rate in later searching period (Dong et al., 2018). The authors applied chaotic mapping function to mitigate the mentioned problems of CSA. The model proposed improve the forecasting capability of SVR algorithm.

Li et al., (2018) discovered that the Fruitfly Optimization Algorithm (FOA) had limitations, including premature convergence and a high likelihood of getting stuck in local optima. To address these issues, the authors proposed enhancements to FOA using Quantum Computing Mechanism (QCM) and a cat chaotic mapping function. QCM was employed to improve the searching ability of FOA and prevent premature convergence. The cat chaotic mapping function was used to assist the algorithm in escaping local optima when population diversity is low. The authors' optimised model demonstrated improvement based on MAE, MAPE, and RMSE as statistical metrics against compared techniques.

Chou & Truong, (2019) conducted study by employing Support Vector Regression (SVR) for accurate forecasting of the exchange rate between the Canadian dollar and the United States dollar (USD). To enhance the performance of SVR, the authors utilized an enhanced firefly algorithm (FA) that was tailored specifically for parameter optimization, employing a sliding-window technique. The authors use Gauss/mouse mapping and logistic mapping for tuning attractiveness of FA and population initialisation activities respectively. Also, the authors us Lévy flight and Adaptive Inertia Weight (AIW) for enhancement of local search and local exploitation cum global exploration capabilities respectively. The model proposed was able to record higher accuracy based on RMSE, MAE and MAPE as statistical metrics for evaluation.

Chou & Pham, (2017) forecast the scour depth effect caused by flowing water against bridge. The authors use SVR as forecasting algorithm optimised with Firefly algorithm. The authors also use chaotic map function for effective random initialisation and Lévy flight to enhance local search. SAFCAF as the developed model shows significant accuracy against benchmarked algorithms.

Tran & Hoang, (2017) used Flower Pollination Algorithm (FPA) to optimise SVR parameters for forecasting algal colony growth on façade structures. They enhanced the FPA's search functionality by incorporating Lévy flight. The resulting LSVR-FPA model outperformed several statistical and machine learning based benchmarks in terms of accuracy, as measured by RMSE and R^2 metrics.

Verma et al., (2017) conducted a study on optimising Support Vector Regression (SVR) for predicting cement compressive strength using multivariate parameters. The authors employed Particle Swarm Optimization (PSO) and Symbiotic Organism Search (SOS) as optimization algorithms for SVR. Their models demonstrated superior accuracy compared to benchmark models such as ANN, RVM, and GPR. Various metrics, including, MSE, MAE, and MAPE were used to evaluate the performance of the models.

Though, Swarm Intelligence methods have proven to be an effective means of parameter optimisation for machine learning algorithms as mentioned in the literature. However, these methods are prone to be trapped in local optima, most of times converged prematurely or takes longer time to converge (Chou & Pham, 2017; Chou & Truong, 2019; Dong et al., 2018; Li et al., 2018). Table 2.5 present the summary of different swarm intelligence algorithms used to optimise SVR algorithm with RBF kernel.

Table 2. 3

Swarm Intelligence based algorithms for SVR parameter optimisation

Sno	Year	Application Area	Hybrid Algorithm
1	Zhang et al., (2019)	Electricity Consumption	ABC
2	Dong et al., (2018)	Household electric demand	Cuckoo search
3	Li et al., (2018)	Grid load Forecast	Firefly
4	Chou & Pham, (2017)	Scour depth forecast	Firefly
5	Tran & Hoang, (2017)	Algal growth forecast	Flower pollination algorithm
6	Verma et al., (2017)	Forecasting of Cement compressive strength	PSO and SOS
7	Mahmoudi et al., (2016)	Forecasting of water quality	Shuffled frog leaping algo (SFLA)

Swarm Intelligence (SI) has been proven to be an effective method of parameter optimisation for machine learning algorithms as demonstrated in the literature. However, these methods exhibited inherent weaknesses of having tendency to be trapped in local optima, slow and premature convergence (Chou & Pham, 2017; Chou & Truong, 2019; Dong et al., 2018; Li et al., 2018) of which African Buffalo Optimisation algorithm is part of. Hence the need for exploring other techniques to mitigate such weaknesses of SI based techniques. ABO algorithm is one of such SI based methods used for optimisation process in machine learning domain.

2.4 African Buffalo Optimisation Algorithm

The African Buffalo Optimisation (ABO) algorithm, developed by Odili, belongs to the class of swarm intelligence (Odili et al., 2015). It models the foraging and defending behaviour of African buffaloes, which exhibit unique features such as extensive memory capacity, communal lifestyle, and democratic decision-making lifestyle (Ghosh, 2022; Ullah Khan et al., 2021; Vaza et al., 2022). These animals communicate danger and safety using the sounds *"waaa"* and *"maaa"* respectively, which are mapped to the algorithm's organisational lifestyle characteristics (Odili et al., 2016; Odili, Kahar, Noraziah, et al., 2017).

The ABO algorithm utilises parameters such as *"waaa"* sound denoted by w_k, *"maaa"* sound denoted by mk, and learning parameters denoted by l_1 and l_2. It also involves global best (bg_{max}) and personal best ($bp_{max(k)}$) positions. The algorithm follows two equations: the democratic equation (Eqn. 2.8) and the location update equation (Eqn. 2.9). Algorithm 2.1 outlines the basic flow of the ABO algorithm. It subtracts the *"waaa"* value (wk) from the maximum vector (bg_{max} and $bp_{max(k)}$), which is then multiplied by the learning parameters (l_1 and l_2). While λ is a variable that determines the time interval over movement of buffalo and generally fixed to 1 (Alweshah et al., 2022; Barnwal et al., 2023; Kesavan et al., 2022; Sushma et al., 2022).

The *"maaa"* value (m_k) indicates that the herds should remain in that location and continue grazing. The exploitation and exploration stages of the ABO algorithm are represented by Eqn. (2.6) and Eqn. (2.7) respectively. The complete basic ABO algorithm is presented in Algorithm 2.1 (Odili et al., 2015).

$$m_{k+1} = m_k + l_1(bg_{max} - w_k) + l_2\left(bp_{max(k)} - w_k\right) \tag{2.6}$$

$$w_{k+1} = \frac{(w_k + m_k)}{\lambda} \tag{2.7}$$

Algorithm 2.1: African Buffalo Optimisation

Step 1: Random initialisation of buffaloes in search space

Step 2: Updating the exploitation behaviour using equation 2.8

Step 3: Update the individual location of buffalo using equation 2.9

Step 4: If equation 3.18 and 3.19 are updating, continue to step 5, else go to step 1

Step 5: If stopping criteria* is reached go to step 6, otherwise go to step 2

Step 6: Output the best result.

*where the stopping criteria can be either the maximum number of iterations is reached, or when the improvement in the fitness value becomes negligible over consecutive iterations.

2.4.1 African Buffalo Optimisation Algorithm in Literature

ABO as metaheuristic-based optimisation algorithm has been compared with other meta-heuristics algorithms (Odili, Kahar, Noraziah, et al., 2017) to establish its performance. The algorithm has also been used in literature for solving various optimisation problems in different domains such as collaborative team formation in social network (El-Ashmawi, 2018), symmetrical and asymmetrical problem of travelling salesman (Odili et al., 2016; Odili & Mohmad Kahar, 2016), spatial modeling of Prey-Predator based cellular automata (Palyulin et al., 2014), numerical function optimisation (Odili & Kahar, 2015), Proportional-Integral-Derivative (PID) controller parameter tuning (Zhang et al., 2018) and determining the best biodiversity area for conservation with constrained budget (Almonacid et al., 2017).

Furthermore, , the algorithm was compared with other well established nature inspired algorithms. Results obtained from the study demonstrated that ABO has a better performance than genetic algorithm (GA), honey-bee mating optimisation (HBMO), ant colony optimisation (ACO) and simulated annealing (SI) and many other metaheuristic algorithms (Odili, Kahar, Noraziah, et al., 2017).

The ABO algorithm has demonstrated notably superior performance compared to Genetic Algorithm (GA) and an enhanced version of GA in various applications. In an evaluation against numerical functions and the tuning of PID controller parameters, the ABO algorithm exhibited remarkable success, surpassing the performance of both GA and the enhanced GA Odili, Kahar, & Noraziah, (2017). Furthermore, when applied to a metaheuristic-based simulation of a dynamic prey-predator model using cellular automata, the ABO algorithm exhibited superior performance (Almonacid, 2017).

In a related study by (Igiri, Singh, & Bhargava, 2019a), the authors further enhanced the ABO algorithm by improving its population initialization and exploration process using logistic-map based chaotic function, and Mantegna-based Levy flight function respectively. This improvement aimed to enhance the algorithm's efficiency and effectiveness in finding optimal solutions. The authors' enhancements contribute to the continuous refinement and advancement of the ABO algorithm for solving complex optimization problems.

Jiang et al., (2020) Improved several aspects of ABO algorithm within domain Scheduling Problem (Energy consumption as the considered factor). The authors improved on Population initialisation, Exploitation and Exploration part of ABO algorithm to produce Improved African Buffalo Optimisation (IABO) algorithm. The improvement based on each section is as follows:

The authors use three different methods (Global selection, Local selection, and Random selection) for population initialisation in ABO. The methods were applied at random at each iteration to generate locations of buffaloes in search space. Similarly**,** the authors introduce three distinct different age-based mechanisms for the buffaloes during training. This is to enhance the exploration process of the proposed algorithm.

At the exploitation stage, the authors added a random buffalo location and a randomly value as new learning additional parameter in the exploitation equation to avoid premature convergence

Krisnawati et al., (2020) conducted study on Flow-shop Scheduling Problem (FSP) in muffler production industry. The authors use classical ABO as an optimiser and use Friedman test to determine the performance of the solution produced by ABO and benchmarked algorithms (Hybrid GA, PSO, and CSA) . The result shows that ABO was able to produce optimal solution compared to other algorithms. However, the authors observed that ABO records higher computational time than other benchmarked algorithms

In a study conducted by (Panhalkar & Doye, 2022) , the authors employ ABO algorithm to improve the shortcomings of Decision trees for feature selection on classification task that has been applied on a large dataset . The authors mentioned that Decision Trees are highly instable and prone to overfitting, Based on the application of ABO algorithm the newly developed hybrid ABODT algorithm was able to outperform all benchmarked algorithms namely Antminer-Decision Trees, and Ant Colony- Decision Trees (ACDT) on four out of six (6) real world dataset used to test the algorithm.

In (Mishra, 2022), the authors employ ABO algorithm to optimise Decision trees for Intrusion Detection System. However, the authors enhanced ABO's population initialisation and exploitation mechanism. Population re-initialisation was modified using Discreet Cross-over method where new buffaloes' locations are created based new arbitrary value between Global best and Personal best.

To modify the exploitation process, the authors employ utilisation of random Swap Operator on both l_1 and l_2 parameters. The developed algorithm was tested on Three datasets on feature selection, and sample selection for classification task. The result shows that the developed hybrid algorithm shows superior performance against classical Decision Trees, SVM, ANN, and KNN on all datasets.

Improved African Buffalo Optimization-Based Takagi–Sugeno–Kang Fuzzy PI Controller for Speed Control in BLDC Motor has been proposed by (Subramani et al., 2023). The authors hybridised ABO with Takagi-Sugeno-Kang (TSKF) fuzzy algorithm to control the speed of BLDC motor. However, upon close inspection, the authors utilise classical ABO algorithm without any improvement. The developed algorithm was compared with PSO, GSA, CSS, GWO, and WOA as benchmarks. The developed IABO outperforms all the benchmarks in terms of determining optimal values, lower optimisation error rate, while only GSA converge faster than IABO.

In another study, (Sheeba et al., 2023) enhanced the exploration mechanism of classical ABO using Mantegna-based Levy flight for feature selection purpose for Deep learning algorithm. The authors argued that the newly developed Intrusion Detection using Modified Buffalo Optimization Algorithm with Deep Learning (IDMBOA-DL) shows a remarkable classification accuracy of 99.50% better than all benchmarks.

Barnwal et al., (2023) The ABO algorithm was used to design a fitness function based on multiple parameters in order to achieve efficient clustering of sensors nodes for better routing. The ABO was hybridised with Whale month flame optimisation algorithm to achieve the said purpose. The ABO was used to mitigate weak exploitation capability of whale Moth Flame Optimisation algorithm. Before employing the ABO algorithm, the authors utilise Oppositional Based Learning (OBL) method for population initialisation in order to improve the convergence speed of ABO.

The developed IABO algorithm has shown a better performance on all metrics (Throughput, Energy consumption, and Network lifetime) compared to benchmarked algorithms slightly followed by GWO.

Algaphari, 2023 evaluated an enhanced ABO algorithm on Travelling Salesman Problem (TSP). The author enhanced ABO population initialisation, and the speed of buffaloes during exploration with Fuzzy matrices instead of classical gaussian-based random numbers. The proposed Fuzzy-ABO algorithm was tested on several TSP dataset including Berlin52, Ulysses16, and Burma14 benchmarks. Where fuzzified ABO was the overall best by achieving smallest and optimal solution to the TSP problem. The developed algorithm was compared with classical ACO and PSO, of which the Fuzzy-ABO outperform both.

In another study conducted by (Singhal et al., 2023), the authors use classical ABO as multi-objective optimiser for Test Case for Fault tolerance. The authors designed a Multi-objective Test case selection and Prioritization (TCS&P) model where the artificial buffalos correspond to the test cases and the path (the result of the buffalo's search) is marked as the selected/prioritized test suite. The developed ABO_TCS&P algorithm exhibit remarkable performance in comparison to Ant Colony Optimisation (ACO) as the benchmarked algorithm of which ABO performed exceptionally higher than the ACO.

The aforementioned studies collectively highlight the superior performance of the ABO algorithm in comparison to other notable swarm-based optimization algorithms, such as GA, and the ongoing efforts to enhance its capabilities through algorithmic improvements. These findings contribute to the growing body of research on metaheuristic algorithms and their potential applications in various fields.

Table 2.6 presents several application domains where ABO algorithm has been applied, and also presents various effort by researchers to enhance the ABO algorithm at different instances.

Table 2. 4

ABO as an optimisation algorithm

Sno	**Study**	**Domain**	**Improvement on ABO**	**Improved Part**	**Method Used for Improvement**	**Remark**
1	Odili et al., (2016)	Solving Travelling Salesman Problem (TSP)	No	N/A	N/A	Outperform GA, HBM, SA etc
2	Odili et al., (2015)	Numerical Function Evaluation	No	N/A	N/A	Outperform both GA and Improved GA
3	Odili et al., (2017)	PID controller parameter tuning	No	N/A	N/A	Outperform ACO, PSO and BFO
4	Almonacid et. al.,(2017)	Budget constraint maximal covering location	No	N/A	N/A	Perform competitively with other metaheuristic algorithms
6	Chinwe et. al., (2019)	Classical Optimisation Problem	Yes	Exploration and Exploitation	Levy flight (Mantegna) Chaotic function (Logistic map)	Best overall performance against PSO on Sphere. Schaffer, Beale, and Bochachvesky standard optimisation functions.
7	Jiang et, al., (2020)	Job Scheduling Problem	Yes	Population, Exploration, and Exploitation	**Pop**: Global, Local, and Random selection **Explr:** Aging-based re-initialisation mechanism **Explt:** Discreet individual update	Performed better than modified GA

8	Krisnawati et. al., (2020)	Flowshop Scheduling Problem	No	N/A	N/A	Performed better than Hybrid GA, PSO and CSA in terms of accuracy with higher computational time
9	Panhalkar et. al., (2022)	Enhance Decision Tree with ABO	No	N/A	N/A	Performed better than AMDT, ACDT
10	Mishra et. al., (2022)	Intrusion Detection System	Yes	Population Initialisation, and Exploitation	**Pop:** Discreet Cross-Over method **Explt:** Random swap of l_1 and l_2	Performed better than Classical DT, SVM, ANN, and KNN
11	Subramani et. al., (2023)	BLDC Motor Control	No	N/A	N/A	Performed better than FA, PSO, GSA, CSS, GWO and WOA algorithms
12	Sheeba (2023)	Big Data in IoT	Yes	Exploration	Mantegna-based Levy flight	The proposed algorithm performed better than Classical DL, SVM, CNN, LSTM, CNN-LSTM
13	Barnawal et. al., (2023)	Wireless Sensor Network	Yes	Population Initialisation	Opposition-based Learning (OBL)	Better performance than LEECH, HEED, MBC, FRLDG, and GWO on different metrics
14	Algaphari et. al., (2023)	General Optimisation (TSP)	Yes	Population Initialisation and Exploration	Fuzzy matrices for both population and Exploration enhancement	Overall best against PSO and ACO
15	Sighal (2023)	Software Engineering (Test Case for Fault Tolerance)	No	N/A	N/A	ABO performed better than ACO as the benchmarked algorithm

2.4.2 Weaknesses of African Buffalo Optimisation

African Buffalo Optimisation (ABO) algorithm as a metaheuristic algorithm has been proven to be one of the best performing optimisation algorithms (Alweshah et al., 2022). This could be associated with its simple implementation and its fewer number of parameters. The algorithm superiority has been established in previous section in relation to different domains and tasks where it has been used. However, with all the strengths of ABO, as a metaheuristic algorithm it has the inherent weaknesses as follows:

2.4.2.1 Population Generation

One notable weakness of ABO lies in its population initialisation strategy, which can lead to a deficiency in diversity among the initial solutions. ABO typically initialises its particles randomly within the search space. However, this random placement might result in particles congregating in localised regions of the solution space, known as convergence to suboptimal solutions or premature convergence. Consequently, this lack of diversity in the initial population can hinder the algorithm's ability to explore and exploit the broader solution space effectively, potentially limiting its capacity to discover the global optimum (Arif et al., 2022; Jiang, Tianhua-Zhu & Deng, 2020; S. Zhang et al., 2019).

2.4.1.2 Poor Exploration

The conventional ABO (African Buffalo Optimisation) algorithm functions by predefining its learning parameters (lp_1 and lp_2) before initiating the execution phase (Odili et al., 2015). Notably, these two learning parameters govern both the individual and collective optimal states of the buffaloes within the algorithm. However, this pre-defined specification of learning parameters has the potential to compromise the algorithm's proficiency in exploration performance. Particularly, it may impede the algorithm's ability to thoroughly explore novel solution spaces efficiently, thus potentially yielding suboptimal performance outcomes (Peace Igiri et al., 2018).

2.4.2.3 Poor Exploitation

Similarly, the process governing the update of positions is constrained by a pre-established lambda parameter, devoid of inherent variability owing to its a priori configuration (El-Ashmawi, 2018). This characteristic susceptibility can potentially result in the entrapment of the algorithm within local optima or experiencing premature convergence during optimisation process (Igiri, Singh, & Bhargava, 2019a). This necessitated various researchers to find better approach to mitigate the mentioned problems of ABO for it to attain most efficient performance.

2.5 Chaotic Map Function

Chaos is a known character of non-linear systems which can be mathematically defined as a randomness generated by a simple deterministic function (Rezaee Jordehi, 2015). However, the random behaviour of chaotic randomness has better dynamical and statistical properties (Tharwat & Hassanien, 2018). The statistical properties of drawing from gaussian distribution by chaos functions makes chaos to be able to go through all values specified with a given range without repetition. This behaviour enables chaos search to be able to escape from falling into local optimal solution. Various stochastic optimisation problems usually get trapped into local optima; however, research has shown that employing chaotic map function results into enabling such optimisation problem to escape from falling into such problem (Igiri, Singh, & Bhargava, 2019a). Basically, chaotic optimisation can simply refer to utilising sequences generated from chaotic map function instead of random values in an optimisation process. Initial value of chaotic map function highly affects its behaviour and is denoted by x_0. Various chaotic functions exist in literature such as logistic, Tent, Sinusoidal, circle, Sinus, Gauss, Chebyshev, Singer (Igiri et al., 2020, Farah & Belazi, 2018; Zaimoğlu et al., 2023; Zhang et al., 2019). However, Logistic and Tent map functions are the popularly known chaotic function used in literature (Sayed et al., 2017). This study adopts the Tent map function for population initialisation and reformulation of ABO exploitation. This is due to the inherent behaviour of Tent chaotic map function to exhibits a continuum of dynamic behaviours, spanning from predictability to chaos, characterised by strong ergodic uniformity (Dong et al., 2018).

2.5.1 Tent Map function

The method of chaotic mapping is an optimization strategy used to transform the initial data series. This transformation highlights its susceptibility to initial conditions and yields numerous distinct periodic patterns, a phenomenon known as chaotic ergodicity. This approach has been employed in several studies to produce diverse population characteristics throughout the optimisation process, enhancing search patterns and preventing premature convergence (Dong et al., 2018). Tent map function has been defined as in Eqn. (2.8).

$$x_{k+1} = \begin{cases} 2x_k \;\; x & \epsilon\ (0, 0.5) \\ 2(1 - x_k)x & \epsilon (0.5, 1) \end{cases} \tag{2.8}$$

Where x_k represents an iterative value of variable *x* in the *k*th step, and *k* represents number of iteration steps.

2.6 Lévy Probability Distribution

Lévy probability distribution (LPD) is a distribution obtained from Lévy flight function. It is a type of random walk introduced by Paul Lévy in 1937 that has a characteristic of intensive probability in its movement (Dash et al., 2021). The Lévy walk phenomenon describes the diffusion pattern observed in organisms, where their searching behaviour is focused on potential solution locations. The Lévy flight foraging hypothesis suggests that organisms migrate from less-resource to more-resource environments, leading to optimal search strategies (Pang et al., 2018). Animals with high memory capabilities utilize this model to explore their search space effectively. The theory of optimal foraging extends the concept of Lévy flight foraging, proposing that organisms prioritize the search for optimal solution locations rather than engaging in aimless exploration within the search space. Loosely speaking, Lévy flights are random walks whose step length is drawn from a distribution, often in terms of a simple power-law formula $L(\lambda)$ $|\lambda|^{-1-\beta}$where 0 $\beta \leq 2$ is an index. Mathematical representation of a *Lévy* distribution as defined in (Kołodziejczyk & Tarasenko, 2021) is as presented in Eqn. (2.9).

$$L(\lambda,\gamma,\mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}}\frac{1}{(\lambda-\mu)^{\frac{3}{2}}} exp\left[\frac{-\gamma}{2(\lambda-\mu)}\right], 0 < \mu < \lambda < \infty \\ 0 otherwise \end{cases} \tag{2.9}$$

where μ > 0 is a minimum step and γ is a scale parameter.

In terms of implementation, generating random numbers using Lévy flights involves two steps: selecting the flight direction appropriately and generating steps that adhere to the Lévy distribution (Suresh & Lal, 2016). To achieve optimal result, the direction value has to be drawn from a uniform distribution, whereas the generation of steps proves to be challenging. Few methods existed in the literature of achieving effective way of providing the steps in *Lévy* distribution, two (2) most prominent ones are through Mantegna and McCulloch algorithms (Bashath et al., 2022; Gopal Dhal et al., 2015; Ismail et al., 2021; Suresh & Lal, 2016). The McCulloch algorithm (McCulloch

& Pitts, 1943) is adopted for Lévy flight implementation in this study due to its better performance in terms of ability to search wider space, faster convergence speed and better accuracy that supersede Mantegna algorithm (Ismail et al., 2021; Singh & Agarwal, 2022; Soneji & Sanghvi, 2014).

The McCulloch algorithm, developed by McCulloch (Bashath et al., 2022), utilizes an explicit formula to generate random numbers from a Lévy process. This formula involves two independent variables, w and φ, which follow a uniform distribution in the range $\left(\frac{-\pi}{2}, \frac{\pi}{2}\right)$ and a standard exponential distribution, respectively. The equation for generating these random numbers is given by Eqn. (2.10) that returns random values as steps (Bashath et al., 2022; Soneji & Sanghvi, 2014):

$$C\,\frac{N_1 N_2}{D} + \tau\, S_a(c, \beta, \tau) \tag{2.10}$$

Where:

$$N_1 = sin\left[\alpha\varphi + tan^{-1}\left(\beta tan\left(\frac{\alpha\pi}{2}\right)\right)\right]$$

$$N_2 = \left(cos\left[(1-\alpha)\varphi - tan^{-1}\left(\beta tan\left(\frac{\alpha\pi}{2}\right)\right)\right]\right)^{\frac{1}{\alpha}-1}$$

$$D = \left(cos\left[tan^{-1}\left(\beta tan\left(\frac{\alpha\pi}{2}\right)\right)\right]\right)^{\frac{1}{\alpha}} (cos\varphi)^{\frac{1}{\alpha}} w^{\frac{1}{\alpha}-1}$$

The function generates a matrix of random numbers with dimensions *n x m*. It requires certain parameters, including the characteristic exponent (α), skewness parameter (β), scale (c), and location parameter (τ). To prevent potential overflow, the minimum value for α is set to 0.1. If any of the input parameters are outside the valid range, the resulting matrix will contain NaNs. In such instances, the algorithm employs Eqn. (2.11) to handle the computation.

$$x = c\left(\frac{cos\big((1-\alpha)\varphi\big)}{w}\right)^{\frac{1}{\alpha}-1} \frac{sin(\alpha\varphi)}{cos(\varphi)^{\frac{1}{\alpha}}} + \tau \tag{2.11}$$

where two special cases are handled separately:

case $\alpha = 2$: This evaluates to Gaussian case where *x* becomes

$$x = c2\sqrt{wsin(\varphi) + \tau} \tag{2.12}$$

case $\alpha = 1$: This evaluates to Cauchy case, hence x becomes

$$x = ctan(\varphi) + \tau \tag{2.13}$$

Soneji & Sanghvi, (2014) use Lévy-flight to enhance searching behaviour of cuckoo algorithm. The authors use McCulloch and Mantegna algorithms as Lévy's random number generator. The enhanced cuckoo algorithm was used as optimisation algorithm for SVR in design of electric load forecasting. The resulting Lévy-based function has shown that McCulloch-based outperformed Mantegna-based Lévy function in terms of execution time. Hence, shows favourable result when benchmarked with Sphere, Ackley, Dixon and Price, Griewank, Step, Lévy, Generalised Schwfel 2.6, Generalised Rosenbrock, Rastrigin and Weierstrass functions.

In another study, Pang et al., (2018) use Lévy flight to help Evolutionary Programming (EP) escape local optima. They enhance Lévy flight by adapting the function to local fitness landscapes. This enhancement improves the searching process of the EP algorithm. The hybridized EP with adaptive Lévy function (HEP) outperforms other techniques on unimodal functions, including Schwefel, Rastrigin, Ackley, Griewank, Foxholes, and more.

Megala et al., (2016) proposed an adaptive Lévy mutation for improved performance. They argued that the tuning parameter of the Lévy function should depend on the specific problem and that different tuning is needed at each stage of the searching process. The authors used Mantegna's algorithm to generate random numbers for the adaptive Lévy function in the Clonal Selection Algorithm (CSA).

Additionally, the modified Lévy function was assessed against the Sphere, generalized Rastrigin, and Ackley functions to determine its effectiveness in escaping local minima.

As shown in Table 2.5, the application of the Lévy function as a random number generator has been documented in the literature for various heuristic algorithms. This highlights the versatility and wide-ranging utilization of the Lévy function within the field of optimization. The studies listed in the table demonstrate the use of the Lévy function in different contexts, showcasing its applicability in solving diverse optimization problems.

These evaluations and applications contribute to the understanding of the Lévy function's performance characteristics and its potential as a valuable tool in improving the efficiency and effectiveness of heuristic algorithms.

Table 2. 5

Lévy-flight as Random Generator

Authors	**Target Algorithm**	**Lévy-flight Random Generator**	**Benchmarks**
Soneji & Sanghvi, (2014)	Cuckoo Search	McCulloch	Sphere, Ackley, Dixon and Price, Griewank, Step, Lévy, Generalised Schwfel 2.6, Generalised Rosenbrock, Rastrigin and Weierstrass functions

Pang et al., (2018)	Evolutionary Programming (EP)	Standard Random Number	Thirty-nine (39) functions including Schwefels, Ackley, Rastrigin, Shekels, Schaffer, Sinusoidal etc
Y. Peng et al., (2013)	PSO	Gaussian	Sphere, Rosenbrock, Rastrigin, Griewank and Ackley
Megala et al., (2016)	Clonal selection Algorithm (CSA)	Mantegna	Sphere, generalised Rastrigin and Ackley

2.7 Research Gap Discovered

The Support Vector Regression (SVR) algorithm is widely used for regression tasks, but its performance relies heavily on selecting appropriate hyperparameters. However, traditional methods like Grid search and Cross-Validation (CV) have limitations such as extensive search ranges, sensitivity to step size, and high computational requirements. To overcome these limitations, researchers have explored swarm optimization techniques with promising results. Yet, the African Buffalo Optimisation (ABO) algorithm, a recently introduced swarm-based approach, has not been studied in combination with SVR. Therefore, this study aims to fill this research gap by implementing the ABO algorithm to find optimal hyperparameters for SVR, introducing a new approach called SVR-ABO.

Another research gap is in the population initialization of the ABO algorithm. Previous studies have used conventional techniques, but there is a need for innovative approaches to improve optimization performance. This study addresses this gap by proposing SVR-*pop*ABO, an algorithm that focuses on effective population initialization to enhance convergence rates and overall performance.

Additionally, the exploration mechanism of the ABO algorithm is an area for improvement. While Gaussian random and Mantegna-based Levy flight mechanism have been commonly used, this study explores application of McCulloch-based Levy flight as an alternative method. The McCulloch-based Levy flight has been identified as a more efficient way to explore the search space, leading to the development of SVR-*explr*ABO, an algorithm that improves optimization performance through enhanced exploration.

Furthermore, the exploitation mechanism of the ABO algorithm also requires improvement. Previous studies relied on utilising Logistic map function, but this study introduces the use of the Tent map function. The resulting algorithm, SVR-*explt*ABO, shows promise in exploiting the search space more effectively.

In conclusion, addressing the research gaps in SVR optimization using the ABO algorithm contributes to the advancement of population initialization, exploration, and exploitation mechanisms. This study presents new algorithms (SVR-ABO, SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO) that enhance the robustness and efficiency of optimization, ultimately improving the performance of SVR in various domains.

2.8 Summary

In this chapter, a range of forecasting techniques from both statistical and machine learning fields were discussed. However, the literature suggests that statistical techniques may not be suitable for accurate forecasting in cases where there is a presence of non-linear relationships among the features. As a result, researchers have turned to various machine learning techniques to develop more effective and efficient algorithms for forecasting purposes. According to the literature, ANN-based techniques such as RNN and LSTM have shown promising results in various domains.

However, these techniques have certain issues that make them unsuitable for electric load forecasting. The most prominent issues include overfitting and the need for optimization of a large number of parameters. On the other hand, SVR has been found to overcome the problems associated with ANN-based methods. SVR algorithm's performance depends on several factors, including the choice of kernel function, the penalty factor (C), the tube size (ε), and the RBF kernel parameter (γ). These factors play a crucial role in determining the accuracy and effectiveness of the SVR algorithm for electric load forecasting. Finding the optimal parameter values for SVR can be challenging. One approach is to use an optimization algorithm like ABO. However, ABO has drawbacks such as aimless searching and premature convergence. To address these issues and achieve optimal SVR values for efficient and accurate forecasting, there is a need to mitigate the shortcomings of the ABO algorithm.

CHAPTER THREE

RESEARCH METHODOLOGY

This chapter presents the deployed methodology in this study. This study followed the outlined five (5) stages as the research framework as depicted in Figure 3.1 (adapted from (Mustaffa, 2014)) based on the categorisation of major tasks involved viz. data collection and preparation, algorithm design, algorithm development, and evaluation.

The source and description of data used in the study and the data treatment applied on the data are hereby presented. At algorithm design stage, the description of steps involved on how to enhance the SVR, population initialisation, exploration and exploitation ability of ABO are also described in subsequent sections. The final resulting hybrid SVR-eABO algorithm is used on the treated dataset for the ELF forecasting purpose. Figure 3.1 depicts the flow of the process.

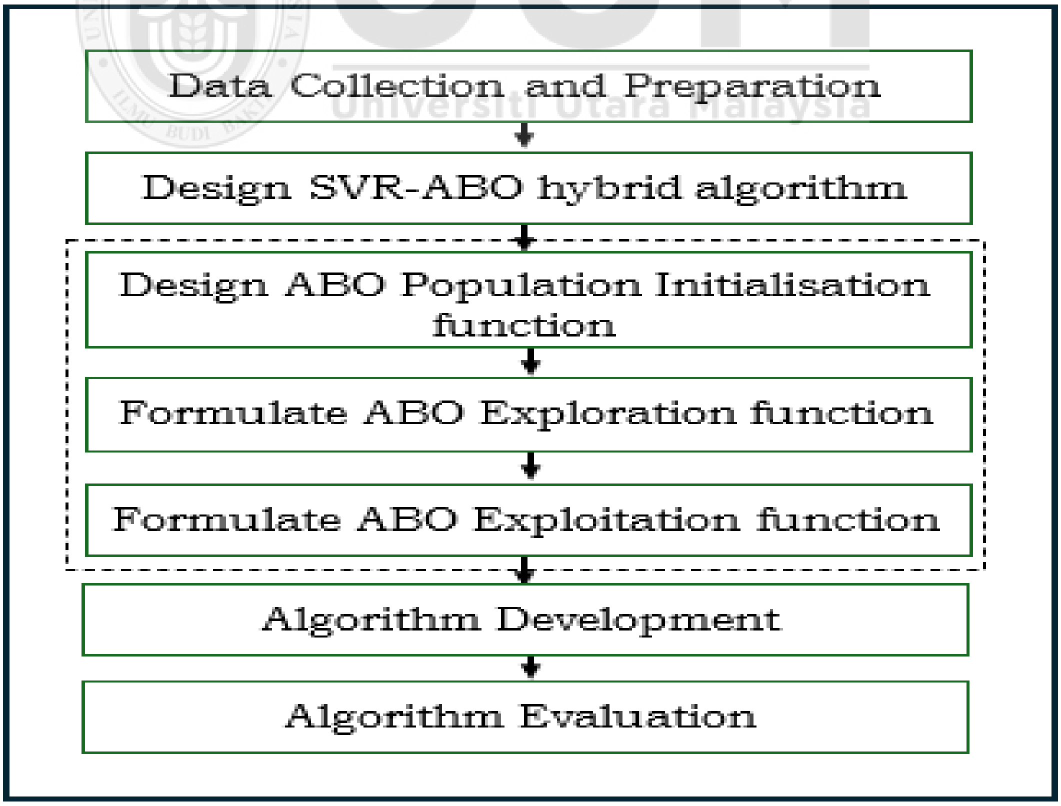

Figure 3.1. Research Process

3.1 Data Collection and Preparation

This research employed four (4) secondary multivariate time-series datasets acquired from the UCI machine learning repository, Mendeley data repository. The datasets are as follows: (a) Individual household electric power consumption dataset. (b) Appliances energy forecasting dataset (c) Turkey electricity load dataset, and (d) Panama electricity load dataset. These datasets were selected based on the suitability for electricity load forecasting as employed in several research in literature (L. M. Candanedo et al., 2017; Gasparin et al., 2022a; Madrid & Antonio, 2021; Sinha et al., 2021; Tutun et al., 2015).

3.1.1 Datasets

The dataset was partitioned into three on 70%, 15%, and 15% ratio for Training, Validation and Testing respectively (Cheung et al., 2018).

3.2.1.1 Individual Household Electric Power Consumption Dataset

The dataset concerning time-series and multivariate household electric power consumption was provided to the UCI repository by Georges Herbrail and Alice Berard of EDF R&D situated in Clamart, France (Hebrail, Georges and Berard, 2012; Sinha et al., 2021) . This dataset was donated to UCI repository on the 30th of August 2012. The description of the data set is as presented in table 3.1.

Table 3.1

Household Dataset

Dataset	Characteristics	Number of Attributes	Attribute Type	No of Instances	Missing values
A	Multivariate, Time-series	9	Real values	2,075,259	Yes

The dataset exhibits a presence of missing values within the measurement data, accounting for approximately 1.25% of the total dataset rows. While all calendar timestamps are recorded, a subset of these timestamps have corresponding measurement values that are absent. In the dataset, the absence of a value is denoted by the lack of information between two consecutive semi-colon attribute separators. As an example, the dataset illustrates the occurrence of missing values specifically on April 28, 2007. To address this issue, we employed a strategy to impute the missing values by substituting them with the mean values of power consumption recorded during the corresponding minutes from other years as performed in (Gasparin et al., 2022a; Mocanu, Nguyen, Gibescu, & Kling, 2016). This is to ensure a more complete, and representative dataset, allowing for a more accurate analysis of the power consumption patterns across time. The description of the nine attributes of the Household dataset is as shown in table 3.2.

Table 3.2

Description Household Dataset Attributes

Sno	Attribute	Description
1	Date	Date in format dd/mm/yyyy
2	Time	Time in format hh:mm:ss
3	**Global_Active_Power**	**Household global minute-averaged active power (in kilowatt)**
4	Global_Reactive_Power	Household global minute reactive power (in kilowatt)
5	Voltage	Minute-averaged voltage (in volt)
6	Global_Intensity	Household global minute-averaged current intensity (in ampere)

7	Sub_metering_1	Energy sub_metering No. 1 (in watt-hour of active energy) *
8	Sub_metering_2	Energy sub_metering No. 2 (in watt-hour of active energy) **
9	Sub_metering_3	Energy sub_metering No. 3 (in watt-hour of active energy) ***

*: Corresponds to kitchen, containing mainly a dishwasher, an oven and a microwave
**: Corresponds to laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light
***: Corresponds to an electric water-heater and an air-conditioner.

The target value that was forecasted from this dataset is the global active power. It comprises of sub_metering_1, sub_metering_2, sub_metering_3 and the remaining difference obtained.

3.2.1.2 Turkey Electricity Consumption dataset

This dataset represents a multivariate monthly record of electricity consumption sourced from Turkey and was released in October 2016 (Tutun, 2016), encompassing twelve (12) attributes. The dataset spans from January 1976 to December 2010, comprising a total of four hundred and twenty (420) records. This dataset was contributed to the Mendeley repository by Tutun Salih and has been notably employed in scholarly works concerning electricity consumption analysis (Tutun et al., 2015). A comprehensive depiction of the dataset's attributes can be found in Table 3.3.

Table 3.3

Description of Turkey Electricity consumption Dataset Attributes

Characteristics	**Number of Attributes**	**Attribute Type**	**No of Instances**	**Missing values**
Multivariate	12	Real	420	No

Description of Turkey electricity consumption attributes is as shown in table 3.4.

Table 3.4

Description of Turkey Electricity consumption Dataset Attributes

Sno	Attribute	Description
1.	Date	Time Stamp (monthly)
2.	Gross Income	Amount of money at people disposal in the current month
3.	Population	Number of present living people
4.	Load	Electricity load (MWh)
5.	Immediate load	Immediate National electricity load (MWh)
6.	Import	Import recorded for the month
7.	Export	Export recorded for the month
8.	Gross production	Total production for the month
9.	Transmitted energy	Amount of electricity transmitted
10.	**Net electricity Consumption**	**Electricity consumed**
11.	T.C electricity Consumption	Electricity demand
12.	Lost electricity	Electricity lost

The target value used in the forecast is attribute "Net electricity consumption, while other attributes serve as features used by the developed model for making forecast

3.2.1.3 Appliances Energy Forecasting Dataset

This dataset represents a collection of energy consumption data for a residential house in Belgium. It is characterised by being a time series dataset with multiple variables. The data was gathered at intervals of ten (10) minutes, spanning over a duration of four and a half months.

There are twenty-nine (29) distinct features in the dataset. These features pertain to various aspects such as temperature and humidity, both of which were sourced from the closest weather station, specifically Chievres Airport in Belgium.

The dataset was initially utilised in a study conducted by (L. Candanedo, 2017), contributed to the UCI repository on February 15, 2017. Additional details regarding the dataset's characteristics and attributes can be found in Table 3.5.

Table 3.5

Appliances Energy Forecasting Dataset

Dataset	**Characteristics**	**Number of Attributes**	**Attribute Type**	**No of Instances**	**Missing values**
B	Multivariate, Time-series	29	Real	19,735	No

The description of the twenty-nine (29) attributes of the Appliances dataset is as shown in table 3.6.

Table 3.6

Description of Appliances Energy Forecasting Dataset Attributes

Sno	**Attribute**	**Description**
1	Date	Year-Month-Day Hour: Minute: Second
2	**Appliances**	**Energy use of appliances (Wh)**
3	**Lights**	**Energy use of light fixtures in the house (Wh)**
4	T1	Temperature in kitchen area (^{o}C)
5	RH_1	Humidity in kitchen area
6	T2	Temperature in living room area (^{o}C)
7	RH_2	Humidity in living room area
8	T3	Temperature in laundry room (^{o}C)
9	RH_3	Humidity in laundry room
10	T4	Temperature in office room (^{o}C)

11	RH_4	Humidity in office room (%)
12	T5	Temperature in bathroom (^{o}C)
13	RH_5	Humidity in bathroom (%)
14	T6	Temperature outside the building (north side) (^{o}C)
15	RH_6	Humidity outside the building (north side) (%)
16	T7	Temperature in Ironing room (^{o}C)
17	RH_7	Humidity in Ironing room (%)
18	T8	Temperature in teenager room 2 (^{o}C)
19	RH_8	Humidity in teenager room 2 (%)
20	T9	Temperature in parent's room (^{o}C)
21	RH_9	Humidity in parent's room (%)
22	T0	Outside Temperature from Chievres airport weather station (%)
23	Pressure	Outside pressure from Chievres airport weather station (%)
24	RH_Out	Outside humidity from Chievres airport weather station (mm Hg)
25	Wind speed	Wind speed from Chievres airport weather station (m/s)
26	Visibility	Visibility readings from Chievres airport weather station (km)
27	Dew Point	Dew point readings from Chievras airport weather station (A^{o}C)
28	RV1	Random variable 1 (non-dimensional)
29	RV2	Random variable 2 (non-dimensional)

The target value of the dataset is the summation of power consumed by appliances and lights, while other values served as the features to assist in determining the target value.

3.2.1.4 Panama Electricity dataset

This is an hourly multivariate dataset comprising of electricity load data along with timestamp as index, temperature, wind, precipitation, and humidity as weather variables obtained from various sources in Panama. The dataset was obtained over period of five years from 2015 to 2020. The dataset has a total of forty-eight thousand, and forty-eight records (48,048) with no missing values. It is a publicly available dataset used in literature for short-term electricity load forecast (Madrid & Antonio, 2021). The description of the dataset is as presented in table 3.7.

Table 3.7

Panama electricity load dataset

Dataset	**Characteristics**	**Number of Attributes**	**Attribute Type**	**No of Instances**	**Missing values**
C	Multivariate	7	Real and binary	49,048	No

The description of the seven (7) features of the Panama dataset are as shown in table 3.8.

Table 3.8

Description of Panama dataset attributes

Sno	**Attribute**	**Description**
1	**National load**	**National electricity load (MWh)**
2	Holiday	Holiday period (binary)
3	School	School period (binary)
4	Temp	Air temperature (^oC)
5	Hum	Specific humidity (%)
6	Wind	Wind speed (m/s)
7	Precipitation	Water droplet in air (l/m^2)

In this dataset, the National load is the target value that has been forecasted from the dataset, while other attributes serve as features used by the developed model for making forecast.

3.2 Data Pre-processing

This phase describes operations performed on the datasets before building the models based on developed algorithms. A comprehensive evaluation was conducted on each dataset to discern the extent of linearity exhibited between the features and the target value. Furthermore, normalisation was applied on each feature to mitigating prospective bias stemming from the features during phase of model training.

Household dataset is the only dataset with missing values. Nearly 1.25% of observations are missing. The missing values were replaced with recorded data from the corresponding time on the previous day, similar process was performed literature (Gasparin et al., 2022b)

3.2.1 Test for Non-Linearity

The main reason behind selecting SVR algorithm instead of other classical statistical time-series based forecasting models is due to the assumed non-linearity properties of the targeted dataset. Hence there is need to ascertain the non-linearity or otherwise of the dataset that will be use in this study. This study uses BDS test in Eviews Statistical software for linearity test on the targeted datasets (Gerolimetto & Bisaglia, 2014; Lim et al., 2005; M. O., 2015; Skare et al., 2019). The result of the test on each dataset proved that there is absence of linearity in the dataset. The results are presented in appendix A.

3.2.2 Data Normalisation

The quality of data fed into machine learning algorithm has a direct influence on the quality of the produced result. Hence, systematic conversion of data to a more standard format is a data processing task that cannot be over emphasised. The presence of difference in magnitude in our datasets, if not addressed, can lead to difficulty of learning by the employed algorithm. Therefore, to ensure the elimination of training bias by the algorithm, the data values of the datasets were normalised by using Decimal Scaling Normalization DSN (Mustaffa & Yusof, 2011; Pan et al., 2016). The DSN method works by moving the decimal point of the values of an attribute X to its maximum absolute value. The number of decimal points moved depends on the maximum absolute value of the dataset. Normalisation of data computed using normalisation formula represented by Eqn. 3.1 (Pan et al., 2016). where given value in the dataset is normalized to by Eqn 3.1

$$x_i' = \frac{x_i}{10^j} \tag{3.1}$$

Where, x' represents a normalised value of the dataset and j represents a smallest value such that max($|x_i'|$)< 1

The selection of this normalisation method was informed based on the comparative analysis result obtained in literature showing the superiority of DSN over Min-Max and Z-score normalisation methods (Mustaffa & Yusof, 2011). Sample of normalised Household table 3.9 and 3.10.

Table 3.9

Sample of Raw Household dataset

Datetime	GAP	GRP	Volt	GI	Total
12/16/2006	1209.176	34.922	93552.53	5180.8	20152.93
12/17/2006	3390.46	226.006	345725.3	14398.6	56507.67
12/18/2006	2203.826	161.792	347373.6	9247.2	36730.43
12/20/2006	2225.748	160.998	348923.6	9313	37095.8

Table 3.10

Sample of Normalised Household dataset

Datetime	GAP	GRP	Volt	GI	Total
12/16/2006	0.120918	0.034922	0.093553	0.051808	0.201529
12/17/2006	0.339046	0.226006	0.345725	0.143986	0.565077
12/18/2006	0.220383	0.161792	0.347374	0.092472	0.367304
12/19/2006	0.166619	0.150942	0.348479	0.07094	0.277699
12/20/2006	0.222575	0.160998	0.348924	0.09313	0.370958

While sample of both raw and normalised data for Turkey dataset is as presented in table 3.11 and 3.12.

Table 3.11

Sample of Raw Turkey dataset

Load	IL	IG	EG	GP	TE	GD	LE	NEC
2676.8	2701.6	0.3	0	1321.1	1194	1530.309	208.0999	1322.209577
2700.5	2736.9	0.2	0	1139.8	1118.2	1428.754	194.2898	1234.464261
2725.6	2762.4	24.5	0	1262.2	1214.6	1541.605	209.6359	1331.969144

IL, IG,EG, GP, TE, GD, LE, and NEC stand for Immediate Load, Immediate Growth, Export Growth, Growth Production, Transmitted Energy, Gross Demand, Lost Electricity and Net Electricity Consumption respectively.

Table 3.12

Sample of Normalised Turkey dataset

Load	IL	IG	EG	GP	TE	GD	LE	NEC
2676.8	2701.6	0.3	0.0	1321.1	1194.0	1530.3	208.1	1322.2
2700.5	2736.9	0.2	0.0	1139.8	1118.2	1428.8	194.3	1234.5

2725.6	2762.4	24.5	0.0	1262.2	1214.6	1541.6	209.6	1332.0
2664.7	2702.2	32.7	0.0	1219.7	1118.0	1428.7	194.3	1234.5

3.3 Algorithm Design

This section presents the approach used to achieve the stated objectives of this research (see Chapter 1, section 1.6). The process was carried out in two (2) phases. In the first phase, classical ABO algorithm was used to optimise SVR hyperparameters. The resulting algorithm is SVR-ABO. Details of the developed SVR-ABO algorithm is presented in Chapter 4, while in the second phase the procedure for enhancing ABO is presented.

In the beginning, the explanation on procedure for the design of new population initialisation function for ABO algorithm is presented. The resulting algorithm from the enhancement is termed as *pop*ABO. In addition, explanation on method used for formulation of an enhanced exploration function for the ABO algorithm is presented. The resulting algorithm from the enhancement of the exploration process is termed as *explr*ABO. Furthermore, the explanation of method used to enhance the exploitation function of ABO is described. The resulting algorithm from the enhancement of the exploitation process is termed as *explt*ABO. In conclusion, all the mentioned enhancement on ABO algorithm at population initialisation phase, exploration phase and at exploitation phases were incorporated together and produce an enhanced ABO (*e*ABO) algorithm. The *e*ABO algorithm was used to optimise SVR hyperparameters, consequently producing a hybrid algorithm named SVR-eABO.

Details of the population initialisation function *explr*ABO, *explt*ABO, and detailed description of SVR-*e*ABO is presented in Chapter 4.

3.3.1 SVR-ABO Algorithm

The classical ABO algorithm has been used in the SVR-ABO algorithm to automatically tune the SVR parameters in this study. The ABO algorithm generated a result after a series of iterations based on boundary values of the search space and objective function, which was used to set the values of the SVR algorithm parameters.

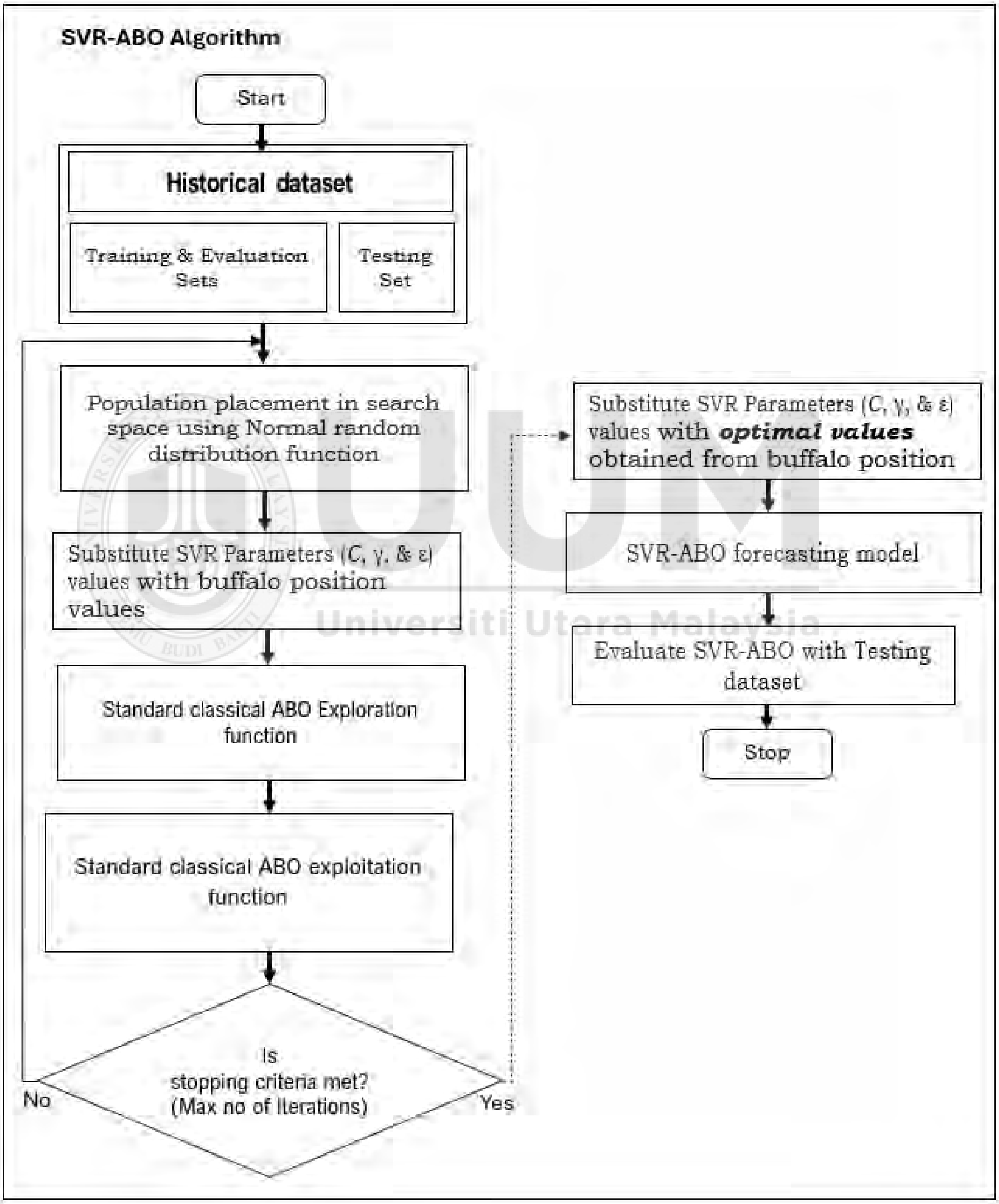

Figure 3.2. SVR-ABO algorithm flow

As presented in figure 3.2 that shows the flow of SVR-ABO algorithm. The ABO algorithm has to be continuously checking during the running time whether there is an improvement in the status of the obtained best global buffalo within the range of specified number of iterations, because stagnation of the buffaloes simply indicates that the buffaloes are trapped in local optima, hence need to restart again. The ABO algorithm supposed to be running until the best value are achieved based on termination criteria, which in this study case is the objective function.

3.3.2 ABO Enhancement

Several steps were explored to enhanced ABO algorithm at various stages of operation. These enhancements were performed due to discovered shortcomings of ABO algorithm as described in literature. The following sub-section provide detailed description of the enhancement performed on ABO at each phase.

3.3.2.1 Population Initialisation

The initialisation stage of the ABO is where the buffaloes are randomly initialised in the search space. Population initialisation is of vital importance and is of high sensitivity in meta-heuristic algorithms, this is because it has tendency to affect the convergence speed and quality of the final solution (S. Zhang et al., 2019). In many nature-inspired optimisation algorithms, the researchers obtain randomness through uniform or Gaussian distribution (Tharwat & Hassanien, 2018). In the event where information about defined boundary of solution is not available, researchers usually resolved to using random initialisation method through chaotic map function (Zhang et al., 2019). Chaotic map functions act like normal random generators but with better dynamic and statistics properties (Tharwat & Hassanien, 2018) The usage of chaotic map in the literature can be broadly categorised into (i) determining global optimal solutions, (ii) generation of chaotic sequences such pseudorandom values, and (iii) providing solution to non-linear equation. In this study, Tent map based chaotic function is used to initialise the buffalo population in the search space (Sayed et al., 2017). Tent map function is mathematically presented as in Eqn. (3.2) (Dong et al., 2018):

$$x_{k+1} = \begin{cases} 2x_k \;\; x & \epsilon\ (0, 0.5) \\ 2(1 - x_k)x & \epsilon(0.5, 1) \end{cases} \tag{3.2}$$

Tent map for chaotic function is being considered as simple, yet effective mapping function with ability to provide better diversity than the one provided by normal randomisation function.

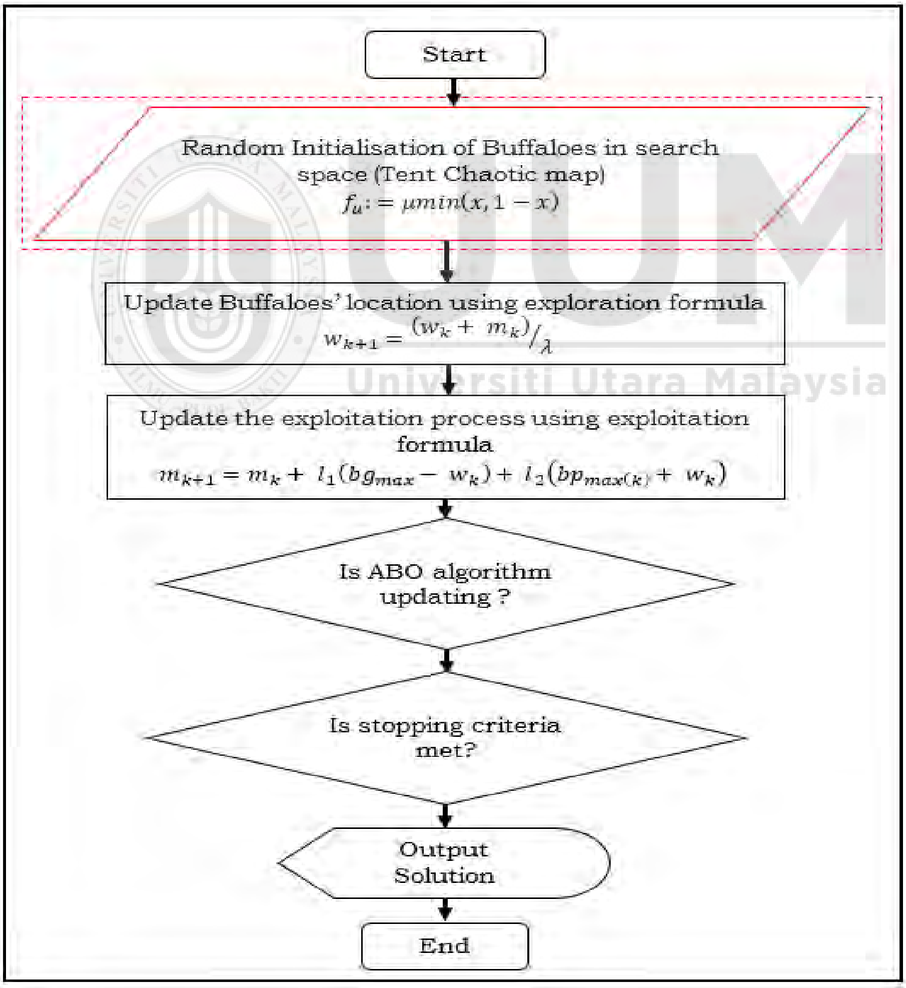

Figure 3.3. Enhancement Population Initialisation phase (*pop*ABO)

Figure 3.3 depict the position of utilising the Tent chaotic function in the enhancement process of ABO algorithm.

3.3.2.2 Exploration Stage Enhancement

The standard African Buffalo Optimisation algorithm exploration process is based on Eqn. (3.3). However, as described in chapter two, section 2.5, this could easily result into an aimless search.

$$w_{k+1} = \frac{(w_k + m_k)}{\lambda} \tag{3.3}$$

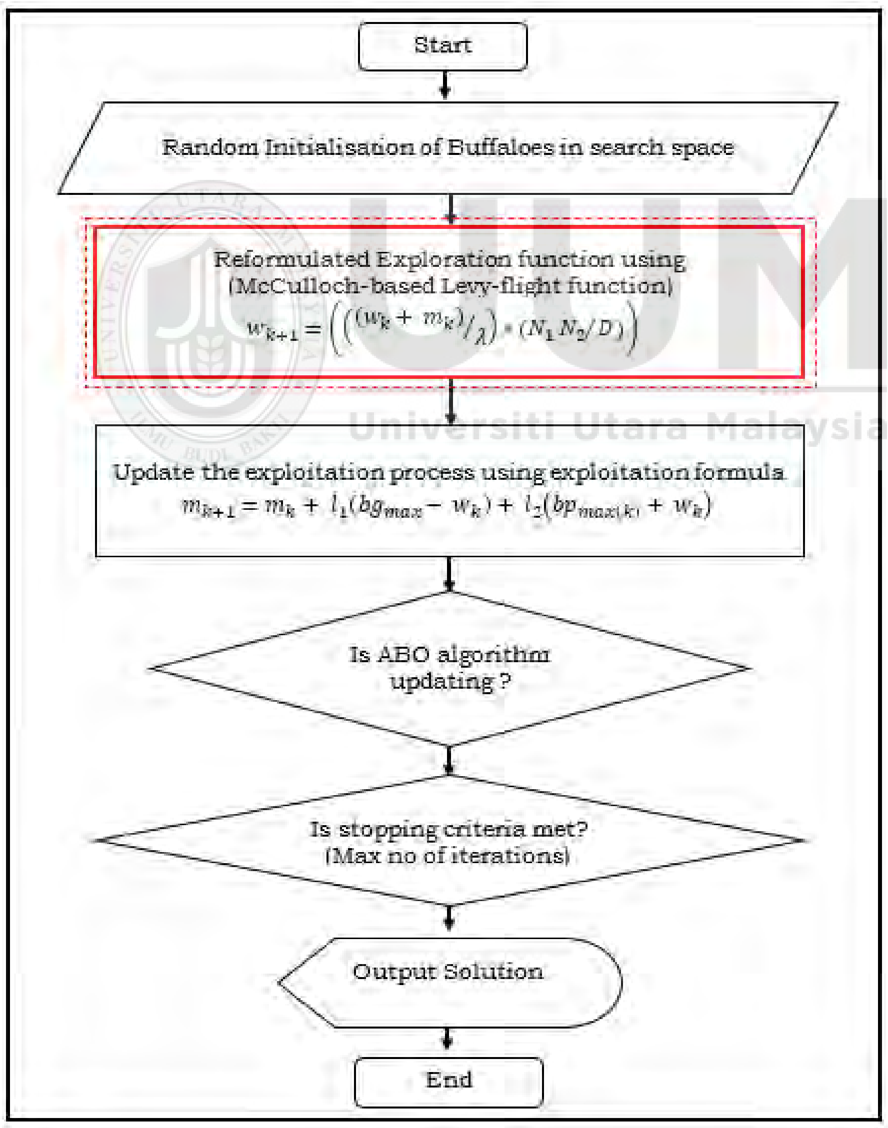

Figure 3.4. Enhanced exploration phase of ABO (*explr*ABO) flowchart

Hence, this research proposes an enhanced exploration process based on Lévy function that will help to guide the exploration process of the buffalo population in the search space. This was achieved by reformulating the updating equation Eqn. (3.5) of ABO with values generated from a Lévy-flight function. The dotted area in figure 3.4 denotes the step where the enhancement in ABO exploration process took place.

3.3.2.3 Exploitation Stage Enhancement

As pointed out in section 2.5.2 that ABO algorithm's exploitation process depends on Eqn. 3.4.

$$m_{k+1} = m_k + l_1(bg_{max} - w_k) + l_2\left(bp_{max(k)} - w_k\right) \tag{3.4}$$

The ABO algorithm's exploitation performance depends on the collective intelligence of the herds by being intelligent enough to know the location of greener pasture which is determine based on previous grazing positions. However, the formulation of the exploitation process of ABO algorithm does not account for this feature (Igiri, Singh, & Bhargava, 2019b). This exposes the algorithm to possible local optima entrapment during exploitation (Igiri et al., 2019b, 2019a; Ben et al., 2017). Therefore, this study uses Tent map function is used as dynamic function for both global and local fitness functions of ABO algorithm as depicted in figure 3.5.

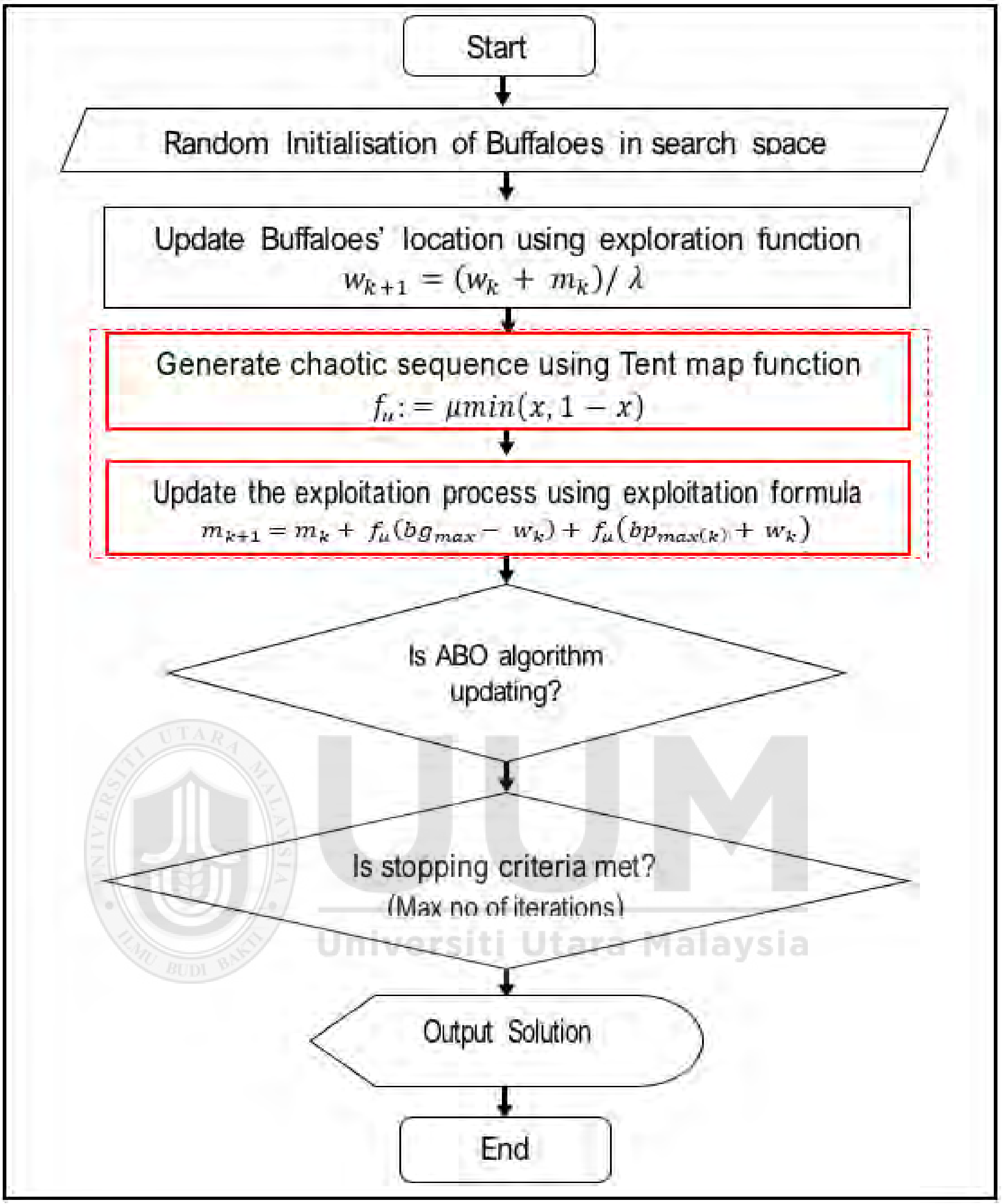

Figure 3.5. Enhanced exploitation phase of ABO (*explt*ABO) flowchart

3.3.3 SVR-eABO Algorithm Flow

The cumulative enhancement ABO algorithm as described in section 3.4.2.1 through section 3.4.2.3 were used to produce an enhanced ABO (*e*ABO) algorithm. The *e*ABO was used to determine optimal parameters of SVR resulting to developing a hybrid SVR-*e*ABO algorithm for forecasting purpose. The overall flow of the SVR-eABO algorithm has been achieved as depicted in figure 3.6.

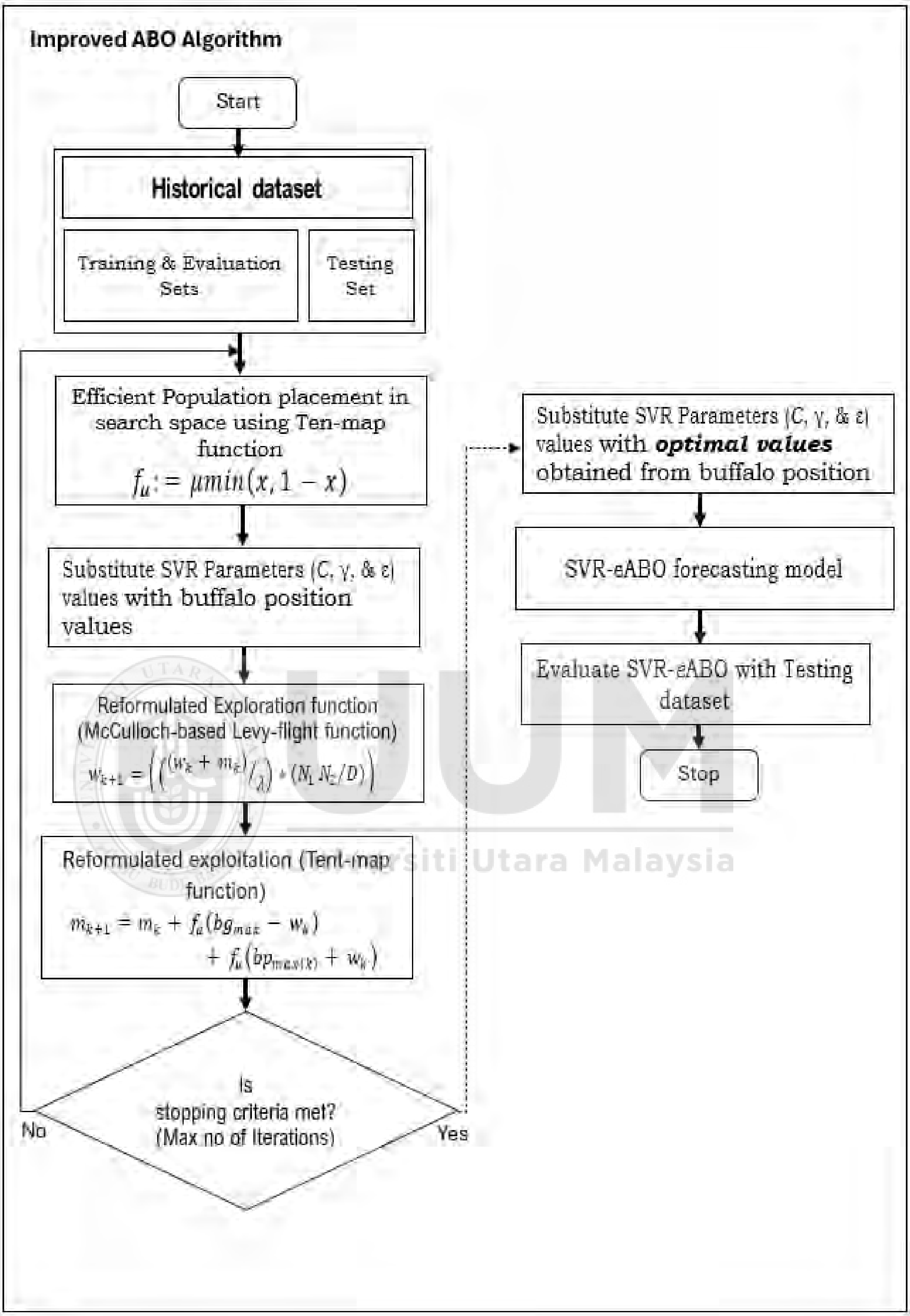

Figure 3.6. Flowchart of SVR-*e*ABO

3.4 Algorithm Development Environment

The proposed algorithm was developed using Python programming language with incorporated Panda, NumPy and Matplotlib modules for dataframe, matrix and visualisation functions respectively. The algorithm was developed on a computer system with specifications as shown in table 3.13.

Table 3.13

System Specification

Sno	Item	Specification
1	Operating system	Fedora 38
2	Linux kernel	4.15.0-46-generic*
2	CPU	Intel Core^{TM} i7-6700HQ @ 2.6 GHz x 4
3	RAM	8 GB
4	HDD	Samsung SSD 512GB

3.5 Evaluation

The performance of the developed algorithms in this study have been evaluated based on several metrics ranging from statistical-based performance metrics (MAPE, MAE RMSE and R^2), Execution time, Convergence speed, Standard optimisation benchmark functions and benchmarked with some selected swarm-based state-of-the-art algorithms.

3.5.1 Performance Metrics

This study uses total of six (6) performance evaluation metrics of which four are purely statistical based metrics. The use of metrics is to determine the performance of the developed algorithms on different dataset used in this study as presented in Chapter 3, (Section 3.2.1). As this study relies on the analysis of time series data, it is imperative to emphasize the criticality of employing a suitable evaluation metric. The selection of an appropriate evaluation metric assumes paramount importance as it serves the crucial purpose of substantiating and justifying the obtained results. Four (4) statistical evaluation metrics have been chosen owing to their suitability for the evaluation of time series data. The four (4) statistical metrics used for performance evaluation in this study are Mean Absolute Percentage (MAPE) as used in (Agga et al., 2022; Halaš et al., 2017; Kristjanpoller & Minutolo, 2018), Root Mean Squared Error (RMSE) as used by (Agga et al., 2022; Chow, 2021), Correlation Coefficient (R^2) as used by (Dieudonné et al., 2023; C. J. Huang & Kuo, 2018; Kari et al., 2018), and Mean Average Error (MAE) as used by (Agga et al., 2022; Dieudonné et al., 2023). The formulas for the MAPE, RSME, R^2, and MAE performance evaluation metrics are presented in equation 3.5, 3.6, 3.7, and 3.8 respectively.

$$MAPE = \frac{1}{N}\left[\sum_{n=1}^{N}\left|\frac{a_i - f_i}{a_i}\right|\right] * 100\% \tag{3.5}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(a_i - f_i)^2}{N}} \tag{3.6}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(a_i - \overline{f}_i\right)^2}{\sum_{i=1}^{N}(a_i - f_i)^2} * 100\% \tag{3.7}$$

$$MAE = \frac{1}{N}\left[\sum_{n=1}^{N}\left|\frac{a_i - f_i}{a_i}\right|\right] \tag{3,8}$$

Where N, represents number of observations, a_i, f_i represent n_{th} individual observed and forecasted values respectively. While $\overline{f}_n$ represents mean of the forecasted data points of the dependant variable.

3.5.2 CPU Execution Time

The CPU execution time as the fifth metric for performance determination was also used on both SVR-ABO and SVR-*e*ABO (Ludwig & Schoene, 2012; Singhal et al., 2023). Similarly, the execution time of SVR-*pop*ABO, SVR-*explr*ABO, and SVR-

*explt*ABO were all evaluated to determine the effect of the enhancement performed on ABO at each corresponding stage.

3.5.3 Percentage Accuracy

The Percentage Accuracy (PA), as the sixth (6) performance metric, has also been employed in this study in order to determine the forecasting accuracy of developed algorithms. The percentage Accuracy shows the degree of making right future forecast in comparison to actual values of test data. The higher the PA value, the better in terms of forecasting model performance . Percentage Accuracy is computed based on the Eqn 3.9 as follows:

$$PA = 100 - \left(\frac{1}{N}\left[\sum_{n=1}^{N}\left|\frac{a_i - f_i}{a_i}\right|\right] * 100\%\right) \tag{3.9}$$

3.5.4 Standard Optimisation Functions

The performance of the enhancement made on population initialisation, exploration and exploitation phases of ABO were evaluated using selected standard optimisation functions found in literature (Bashath et al., 2022; J. S. Chou & Pham, 2017; Jamil & Yang, 2013; Soneji & Sanghvi, 2014). The optimisation functions employed to evaluate the performance of SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO algorithms are as follows: F1 = Sphere, F2 = SumSquares, F3 = Whitley, F4 = Griewank, F5 = Ackley, F6 = Pinter, F7 = Rastrigin, F8 = Schaffer, F9 = Rosenbrock, F10 = Schwefel, F11 = Alpine, F12 = Dixonprice, F13 = Zakharov, F14 = Powell, F15 = Csendes, F16 = Weierstrass. These functions comprise of both unimodal and multimodal meant to test exploitation and exploration capability of an algorithm due to various level and type of challenges posed to optimisation algorithms.

These functions are widely utilised as benchmarks to evaluate the exploration and exploitation abilities of optimization algorithms. The classical functions selected exhibit distinctive properties, with some characterized by multiple local minima (multimodal), which can potentially ensnare a searching algorithm, while others present wide plateaus containing challenging-to-reach single optima (unimodal). An algorithm capable of effectively navigating through multimodal functions demonstrates strong exploration capabilities, whereas one that accurately identifies a single optimum in unimodal functions showcases excellent exploitation capabilities.

The suitability of these functions as optimization benchmarks is well-established in the existing literature due to their intrinsic characteristics and challenges, they pose to searching algorithms (Jamil & Yang, 2013). Table 3.14 provides a comprehensive overview of the mentioned standard benchmarks, including essential details such as their global optima position and value, modality, and corresponding mathematical formulas. While results obtained for population initialisation, exploration and exploitation phases are presented in Chapter 5, (Section 5.6), Chapter 6, (Section 6.6), Chapter 7, (Section 7.6), and Chapter 8, (Section 8.6) respectively.

Table 3.14

Benchmark functions

Sno	Function Name	Mathematical Formula	Modality	Search space	Global Minimum
1	Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$	MM	$[-10,10]$	$0(0,0,\ldots 0)$
2	SumSquares	$f_1 = \sum_{i=1}^{D} i x_i^2$	UM	$[-10,10]$	$0(0,0,\ldots 0)$
3	Whitley	$f_3 = \sum_{i=1}^{D}\sum_{j=1}^{D}\left[\frac{\left(100\left(x_i^2 - x_j\right)^2 + \left(1 - x_j\right)^2\right)^2}{4000}\right]$	MM	$[-100,100]$	$0(0,0,\ldots 0)$

4	Griewank	$f_4 = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	MM	$[-100,100]$	$0(0,0,\dots 0)$
5	Ackley	$f_4 = -20exp\left\{ {}^{-0.2}\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2} \right\} - exp\left\{\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right\} + 20 + e$	MM	$[-35,35]$	$0(0,0,\dots 0)$
6	Pinter	$f_6 = \sum_{i=1}^{D} ix_i^2 \sum_{i=1}^{D} 20isin^2 A + \sum_{i=1}^{D} ilog_{10}(1 + iB^2)$	MM	$[-5.12,5.12]$	$0(0,0,\dots 0)$
7	Rastrigin	$f_7 = \sum_{i=1}^{D} \left[x_i^2 - 10cos(2\pi x_i) + 10\right]$	MM	$[-5.12,5.12]$	$0(0,0,\dots 0)$
8	Schaffer1	$f_8 = 0.5 + \frac{sin^2(x_1^2 + x_2^2)^2 - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}$	UM	$[-100,100]$	$0(0,0,\dots 0)$
9	Rosenbrock	$f_9 = \sum_{i=1}^{D-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	UM	$[-30,30]$	$0(1,1,\dots 1)$
10	Schwefel	$f_{10} = \left(\sum_{i=1}^{D} x_i^2\right)^{\alpha}$	UM	$[-100,100]$	$0(0,0,\dots 0)$
11	Alpine1	$f_{11} = \sum_{i=1}^{D} \lvert x_i sin(x_i) + 0.1x_i \rvert$	MM	$[-10,10]$	$0(0,0,\dots 0)$
12	Dixonprice	$f_{12} = (x_{1-1})^2 + \sum_{i=1}^{D} i\left(2x_i^2 - x_{i-1}\right)^2$	UM	$x_i \epsilon[-10,10], i = 1,2,3,..D$	$0(0,0,\dots 0)$
13	Zakharov	$f_{13} = \sum_{i=1}^{n} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^4$	MM	$x_i \epsilon[-5,5], i = 1,2,3,..D$	$0(0,0,\dots 0)$
14	Powell Sum	$f_{14} = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	UM	$x_i \epsilon[-1,1], i = 1,2,3,..D$	$0(0,0,\dots 0)$
15	Csendes	$f_{15} = \sum_{i=1}^{D} x_i^6 \left(2 + sin\frac{1}{x_i}\right)$	MM	$x_i \epsilon[-1,1], i = 1,2,3,..D$	$0(0,0,\dots 0)$
16	Weierstrass	$f_{16} = \sum_{i=1}^{n} \left[\sum_{k=0}^{kmax} a^k cos\left(2\pi b^k (x_i + 0.5)\right)\right]$	MM	$x_i \epsilon[-0.5,0.5], i = 1,2,3,..D$	$0(0,0,\dots 0)$

Where *MM* and *UM* represent Multimodal and Unimodal respectively.

3.5.5 Benchmarks

The performance of SVR-ABO, SVR-*pop*ABO, SVR-*explr*ABO, SVR-*explt*ABO, and SVR-*e*ABO algorithms have been benchmarked with classical SVR (Smola & Scholkopf, 2004), SVR-ABC, SVR-GA (Xie et al., 2017), SVR-Cuckoo (Dong et al.,

2018) and SVR-PSO (Mohanad et al., 2018). Similarly, the results obtained from. The final developed SVR-*e*ABO algorithm has also been benchmarked with other meta-heuristic based SVR algorithms as follows: SVR-GA and SVR-PSO.

3.6 Summary

This chapter presents the methodology followed to achieve the proposed objectives of this study. Initially, the datasets have been described. Followed by procedures applied to hybridise SVR with classical ABO for SVR hyperparameter optimisation that produced SVR-ABO. Then description of procedures applied for the enhancement of ABO algorithm at population initialisation, exploration, and exploitation phases that produced *e*ABO was presented. Subsequently, the process of hybridising SVR with *e*ABO that produced SVR-*e*ABO was also presented. Finally, the metrics, standard optimisation functions, and benchmarks algorithms used for evaluation at each phase of the algorithm development were presented. Figure 3.7 depicts the flow of the study.

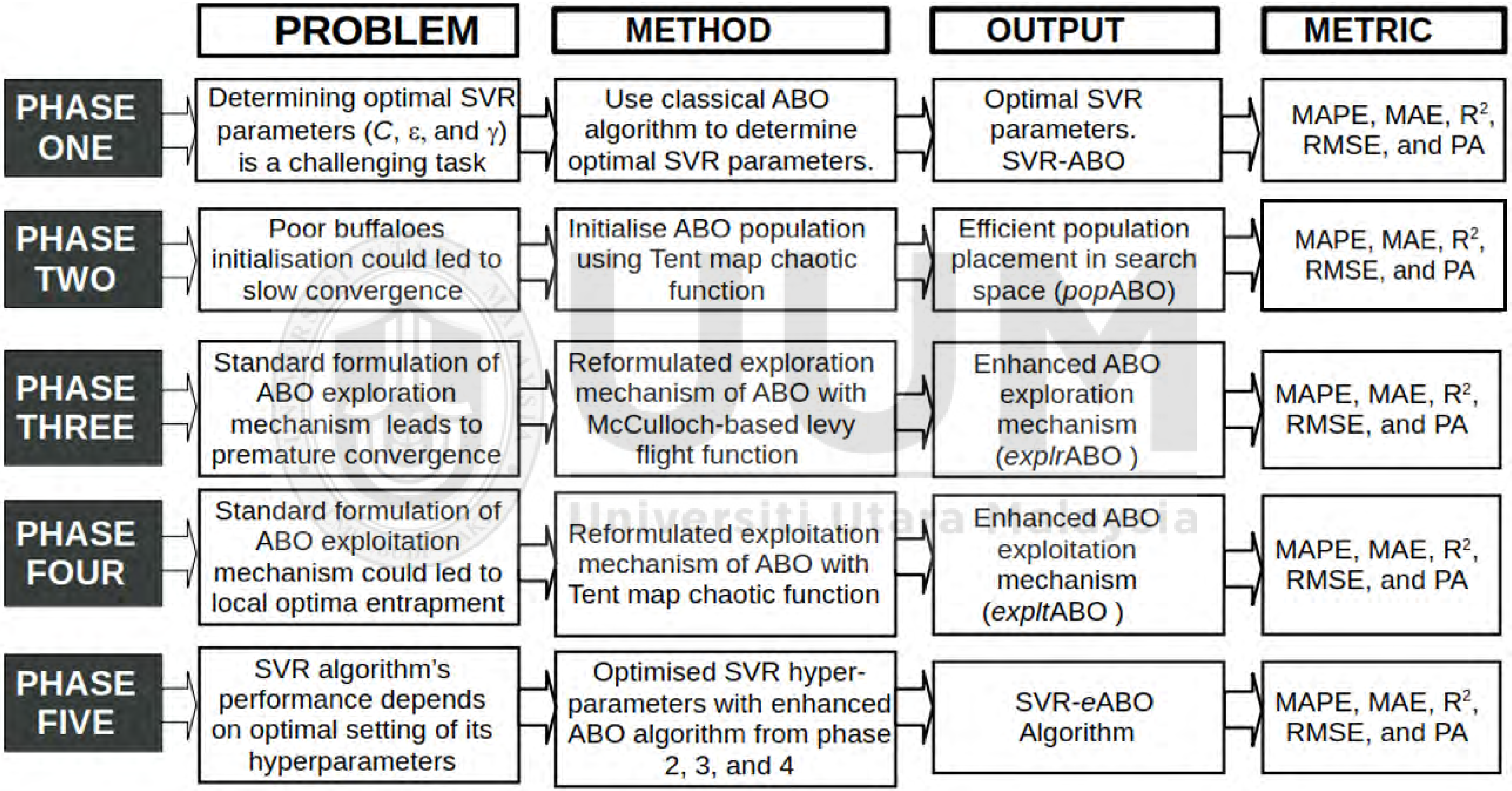

Figure 3.7. General Flow of the Research

CHAPTER FOUR

AN ENHANCED AFRICAN BUFFALO OPTIMISATION ALGORITHM

This chapter presents all the milestones achieved in this study which were defined as objectives in Chapter 1, (Section 1.6). The primary objective of this study as earlier highlighted is to determine optimal hyperparameters of SVR algorithm using an enhanced ABO as an optimising algorithm. However, before several enhancements on ABO algorithm, classical ABO has been used as an optimisation algorithm for SVR. Subsequent sections present detailed explanation of each milestone achieved.

4.1 SVR-ABO algorithm

In this section, we employed the classical ABO algorithm to automatically select hyperparameters for the SVR algorithm. The hybridized SVR-ABO algorithm was evaluated on four datasets mentioned in Chapter 3, (Sec 3.1.1), using regression performance metrics viz: MAE, RMSE, MAPE, and R^2. As already mentioned, the SVR's performance heavily relies on hyperparameter selection, which can be challenging. To address this, we integrated the ABO algorithm with SVR to optimize the hyperparameters. The ABO algorithm's position values were used as potential hyperparameter values, and the MAPE was minimized to determine the optimal values. The SVR-ABO algorithm is presented in Algorithm 4.1.

Algorithm 4.1. SVR-ABO Algorithm

Input: *Training Data, P D, Max_I, l1, l2*
/* *P= Number of individual Buffaloes (population size), D= Problem dimension (SVR control parameters), Max_I = Maximum number of Iterations, l1= Cognitive Learning parameter, l2= Social Learning parameter*/*

Output: *Optimal values for SVR (**C,** γ* and *ε) as Global best buffalo position*

1: **For** buffalo i = 0 to P **do:**

2: Random initialisation of buffalo position vector m_k with three (3) values based on [***C,*** **γ** and ***ε***] ranges using Gaussian distribution

3: Random initialisation of each buffalo movement vector w_k

4: **End For**

5: *Initialise t = 1*

6: **While** (t≠ *Max_I)* **do:**

7: **For** each buffalo i **do:**

8: Calculate fitness_value using SVR regressor

9: **If** buffalo's fitness_value is better than $bp_{max(k)}$

10: Set $bp_{max(k)}$ = buffalo's current fitness

11: **End If**

12: **End For**

13: Set bg_{max} = Best previous buffalo's fitness_value

14: /* Updating each buffalo's movement and position ***/**

15: **For** buffalo i = 0 to P ***do:***

16: **For** dimension d = 0 to D ***do:***

17: $$m_{id}^{t+1} = m_{id}^{t} + l_1(bg_{max} - w_{id}^{t}) + l_2(bp_i^t - w_{id}^{t})$$

$$w_{id}^{t+1} = \frac{(w_{id}^{t} + m_{id}^{t+1})}{\lambda}$$

18: **End For**

19: **End For**

20: *Set t = t + 1*

21: **End While**

22: Evaluate the solution on testing set

23: **Result:** The forecasting values and performance measurement on the testing set

The developed SVR-ABO algorithm has been evaluated on several datasets see Chapter 3, (Section 3.2.1) and the result obtained has been evaluated based on selected regression metrics see Chapter 3, Section 3.6.1.

4.2 An Enhanced Population Initialisation in African Buffalo Optimisation Algorithm

In this section, the ABO algorithm part of SVR-ABO has been enhanced by modifying the population initialization mechanism using the chaotic Tent-map function, resulting in SVR-*pop*ABO. The conventional ABO algorithm's population generation is by using Gaussian-based random numbers which hinders convergence speed, as discussed in Chapter 3, Section 3.3. To overcome this limitation, this study introduced the Tent map function for population generation within the search space as highlighted in Algorithm 4.2. The performance evaluation of the enhanced algorithm utilised four datasets detailed in Chapter 3, Section 3.1.1. The enhanced algorithm has been evaluated using several metrics as explained in Chapter 3, Section 5.1.

Algorithm 4.2. SVR-*pop*ABO Algorithm

Input: *Training Data, P D, Max_I,* l_1, l_2
/* *P= Number of individual Buffaloes (population size), D= Problem dimension (SVR control parameters), Max_I = Maximum number of Iterations,* l_1*= Cognitive Learning parameter,* l_2*= Social Learning parameter**/

Output: *Optimal values for SVR (**C,** **γ** and **ε**) as Global best buffalo position*

For buffalo i = 0 to P **do:**
  Random initialisation of buffalo position vector m_k with three (3) values based on [***C,*** **γ** and ***ε***] ranges using distribution based on **Tent-map** function as follows: $x_{n+1} = \begin{cases} 2x_n x \in [0,0.5] \\ 2(1-x_n)x \in [0,0.5] \end{cases}$
  Random initialisation of each buffalo movement vector w_k
End For
Initialise t = 1
While (t≠ *Max_I)* **do:**
    For each buffalo i **do:**
      Calculate fitness_value using SVR regressor

```
                    If buffalo's fitness_value is better than bp_max(k)
                         Set bp_max(k) = buffalo's current fitness
                    End If
          End For
          Set bg_max = Best previous buffalo's fitness_value
          /* Updating each buffalo's movement and position */
          For buffalo i = 0 to P do:
               For dimension d = 0 to D do:
                    m_id^(t+1) = m_id^t + l_1(bg_max − w_id^t) + l_2(bp_i^t − w_id^t)
                    w_id^(t+1) = (w_id^t + m_id^(t+1)) / λ
               End For
          End For
          Set t = t + 1
End While
Evaluate the solution on testing set
Result: The forecasting values and performance measurement on the testing
set
```


4.3 An Enhanced Exploration in African Buffalo Optimisation Algorithm

In this section, the exploration part of the ABO algorithm has been enhanced by incorporating Lévy flight. The standard exploration process in the ABO algorithm, as described in Section 3.3.4, is based on Equation (4.1):

$$w_{k+1} = \frac{(w_k + m_k)}{\lambda} \tag{4.1}$$

However, this approach can result in an undirected search and could led to premature convergence due to inefficient in a wider search space as elicited Chapter 1, Section 1.4. To address this limitation and prevent early convergence and assist the buffalo population to escape local minima during exploration, a McCulloch-based Lévy flight function was used to enhance the exploration mechanism. The exploration process is modified according to Equation (4.2):

$$w_{k+1} = \frac{(w_k + m_k)}{\lambda} \otimes Levy(\lambda) \tag{4.2}$$

Here, Levy(λ) represents random walks with step sizes following the Lévy distribution, which is defined as:

$$Levy(\lambda) = t^{-\lambda}; 1 < \lambda \leq 3 \tag{4.3}$$

The non-linear relationship in the variance of a Lévy flight enables more effective exploration of large unknown search spaces compared to models with a linear relationship. The iterative process continues until the global optimum is reached, thereby avoiding the problem of getting trapped in local optima that often occurs in the ABO algorithm. The modified algorithm (termed as SVR-*explr*ABO) is presented in Algorithm 4.3.

Algorithm 4.3: SVR-*explr*ABO Algorithm

Input: Training Data, P, D, Max_I, lp_1, lp_2
/* P= Number of individual Buffaloes (population size), D= Problem dimension (SVR control parameters), Max_I = Maximum number of Iterations, i=present Iteration, lp_1= Cognitive learning parameter, lp_2= Social Learning parameter, $\xi = Lévyflight$ */

Output: Optimal values for SVR (**C, γ** and **ε**) as Global best buffalo position

For buffalo i = 0 to P **do:**
    Random initialisation of buffalo position vector m_k with three (3) values based on [***C*, γ** and ***ε***] ranges using normal gaussian distribution
    Random initialisation of each buffalo movement vector w_k
End For
Initialise t = 1
While (t≠ *Max_I)* **do:**
    For each buffalo i **do:**
        Calculate fitness_value using SVR regressor
            If buffalo's fitness_value is better than $bp_{max(k)}$
                Set $bp_{max(k)}$ = buffalo's current fitness
            End If
    End For
    Set bg_{max} = Best previous buffalo's fitness_value
    /* Updating each buffalo's movement and position ***/**
    For buffalo i = 0 to P ***do:***

For dimension d = 0 to D ***do:***

$$m_{id}^{t+1} = m_{id}^{t} + l_1 \otimes \boldsymbol{Levy(\lambda)}(bg_{max} - w_{id}^{t}) + l_2(bp_i^t - w_{id}^{t})$$

$$w_{id}^{t+1} = (w_{id}^{t} + m_{id}^{t+1}) + \otimes \boldsymbol{Levy(\lambda)}$$

End For

End For

Set t = t + 1

End While

Evaluate the solution on testing set

Result: The forecasting values and performance measurement on testing set

4.4 An Enhanced Exploitation in ABO

In this section, the exploitation part of the ABO algorithm is enhanced through the introduction of Equation (4.4).

$$f_{\mu} := \mu min(x, 1 - x) \qquad (4.4)$$

The modification aims to maximize the algorithm's exploitation potential, resulting in improved convergence speed and avoidance of local optima entrapment (Tarkhaneh & Shen, 2019). The modified algorithm (i.e SVR-*explt*ABO) is presented below as Algorithm 4.4.

Algorithm 4.4: SVR-*explt*ABO Algorithm

Input: Training Data, P D, Max_I, lp_1, lp_2
/* P= Number of individual Buffaloes (population size), D= Problem dimension (SVR control parameters), Max_I = Maximum number of Iterations, i=present Iteration, lp_1= Cognitive learning parameter, l2= Social Learning parameter, $LL\acute{e}vyflight$ */

Output: Optimal values for SVR (**C, γ** and **ε**) as Global best buffalo position

For buffalo i = 0 to P **do:**

Random initialisation of buffalo position vector m_k with three (3) values based on [***C*, γ** and ***ε***] ranges using normal gaussian distribution

Random initialisation of each buffalo movement vector w_k

End For

Initialise t = 1

While (t≠ *Max_I)* **do:**

For each buffalo i **do:**

Calculate fitness_value using SVR regressor

If buffalo's fitness_value is better than $bp_{max(k)}$

Set $bp_{max(k)}$ = buffalo's current fitness

End If

End For

Set bg_{max} = Best previous buffalo's fitness_value

/* Updating each buffalo's movement and position ***/**

For buffalo i = 0 to P ***do:***

For dimension d = 0 to D ***do:***

$$m_{k+1} = m_k + f_\mu(bg_{max} - w_k) + f_\mu\big(bp_{max(k)} + w_k\big)$$

$$w_{id}^{t+1} = (w_{id}^{t} + m_{id}^{t+1})$$

End For

End For

Set t = t + 1

End While

Evaluate the solution on testing set

Result: Forecasting values and performance measurement on the testing set

4.5 SVR with an Enhanced ABO

In this section, SVR-*e*ABO as hybrid algorithm has been introduced. The algorithm combines the enhanced ABO (*e*ABO) algorithm with the classical SVR algorithm. The *e*ABO algorithm incorporates the enhancements performed on ABO algorithm at different stages of this study (population, exploration, and exploitation). The hybrid SVR-*e*ABO has been evaluated using datasets mentioned in Chapter3 (section 3.2.1) The results obtained shows significant influence in selecting optimal SVR parameters for optimisation purpose. Results are presented in Chapter 5 with detailed analysis and interpretations. The SVR-eABO algorithm is presented in algorithm 4.5.

Algorithm 4.5: SVR-*e*ABO Algorithm

Input*: Training Data, P D, Max_I, l_1, l_2*
/ P= Number of individual Buffaloes (population size), D= Problem dimension (SVR control parameters), Max_I = Maximum number of Iterations, l1= Cognitive Learning parameter, l2= Social Learning parameter*/*

Output: *Optimal values for SVR (**C, γ** and **ε**) as Global best buffalo position*

For buffalo i = 0 to P **do:**

Random initialisation of buffalo position vector m_k with three (3) values based on [***C*, γ** and ε] ranges using distribution based on **Tent-map** function as follows: $x_{n+1} = \begin{cases} 2x_n x \in [0,0.5] \\ 2(1-x_n)x \in [0,0.5] \end{cases}$

Random initialisation of each buffalo movement vector w_k

End For

Initialise t = 1

While (t≠ *Max_I)* **do:**

For each buffalo i **do:**

Calculate fitness_value using SVR regressor

If buffalo's fitness_value is better than $bp_{max(k)}$

Set $bp_{max(k)}$ = buffalo's current fitness

End If

End For

Set bg_{max} = Best previous buffalo's fitness_value

/* Updating each buffalo's movement and position ***/**

For buffalo i = 0 to P ***do:***

For dimension d = 0 to D ***do:***

$m_{id}^{t+1} = m_{id}^{t} + l_1 \otimes Levy(\lambda)(bg_{max} - w_{id}^{t}) + l_2 L.* (bp_i^t - w_{id}^{t})$

$w_{id}^{t+1} = (w_{id}^{t} + m_{id}^{t+1}) + \otimes Levy(\lambda)$

End For

End For

Set t = t + 1

End While

Evaluate the solution on testing set

Result: The forecasting values and performance measurement on testing set

4.6 Summary

In this chapter, the algorithms developed through targeted enhancements to the African Buffalo Optimization (ABO) algorithm were presented in detail. The chapter began with the introduction of the SVR-ABO algorithm, which serves as a foundational integration of Support Vector Regression and ABO algorithm. Subsequent sections elaborated on specific enhancements, including improved population initialization, exploration, and exploitation techniques, each aimed at bolstering the optimization effectiveness of the ABO algorithm. The chapter concluded with the presentation of the SVR with an Enhanced ABO, showcasing the cumulative impact of these modifications on algorithmic performance. This chapter emphasized the structured development of these algorithms, laying the groundwork for the subsequent analysis of their effectiveness on various datasets, which will be explored in the following chapter. The focus on algorithmic enhancements sets a clear context for understanding how these advancements contribute to improved forecasting modeling and optimization in machine learning applications.

CHAPTER FIVE
DISCUSSION AND ANALYSIS

This chapter transitioned from the development of the enhanced algorithms to their practical application and evaluation. The algorithms introduced in the previous chapter are rigorously tested on several datasets, allowing to assess their performance in real-world scenarios. Utilizing the metrics outlined in Chapter 3, section 3.5, each of developed algorithm effectiveness has been systematically evaluated thereby providing a comprehensive analysis of each algorithm's forecasting capabilities. The results obtained from these experiments are thoroughly discussed, highlighting the strengths and weaknesses of the enhanced algorithms in various contexts. This chapter aims to offer valuable insights into how these advancements translate into measurable improvements in performance, thereby establishing a clear link between algorithmic enhancements and their impact on forecasting modeling outcomes.

5.1 Household dataset

In this section, the performance of various improved algorithms was evaluated using a household dataset. Additionally, the performance of the final algorithm (SVR-*e*ABO) was compared to selected benchmarks, as discussed in Chapter 3, section 3.6.5, using the same dataset. The results obtained from these evaluations are presented in Table 5.1 and 5.2, as shown in sections 5.1.1 and 5.1.2 respectively.

5.1.1 Comparison Between Algorithms on Household Dataset

This section presents the results obtained from evaluating the five algorithms developed in this study on Household dataset. The obtained results of the evaluations are presented below in table 5.1, while the interpretation and analysis of the result followed in subsequent sub-sections.

Table 5.1

Comparative performance of algorithms on Household Dataset

	SVR-ABO	**SVR-*pop*ABO**	**SVR-*explr*ABO**	**SVR-*explt*ABO**	**SVR-*e*ABO**
C	4077.5628	1634.4331	1534.8185	1994.0814	1949.6717
Epsilon	0.2127	0.1897	0.1409	0.12090	0.0979
Gamma	0.0101	0.0032	0.0082	0.00350	0.0018
RMSE	**679.2352**	666.4441	495.2717	390.9594	327.4449
MAPE	**3.0544**	3.1949	**2.2088**	1.7844	**1.4924**
MAE	514.9804	521.2221	352.1861	273.9614	**239.2793**
R^2	0.9941	0.9943	0.9969	0.9980	**0.9986**
PA (%)	**96.9456**	**96.8051**	**97.7912**	**98.2156**	**98.5076**
CPU Time	**19.0383**	16.3259	23.1244	19.4708	24.4815

5.1.1.1 Root Mean Square Error (RMSE)

The RMSE measures the average deviation between the predicted values and the actual values. It provides an overall assessment of the accuracy of the algorithms. Among the algorithms, SVR-eABO achieved the lowest RMSE (327.4449), indicating the smallest average deviation from the actual values. SVR-*explt*ABO (390.9594), SVR-explrABO (495.2717), SVR-popABO (666.4441), and SVR-ABO (679.2352) follow with progressively higher RMSE values. Figure 5.1 depict a visual comparison of RMSE values of the developed algorithms.

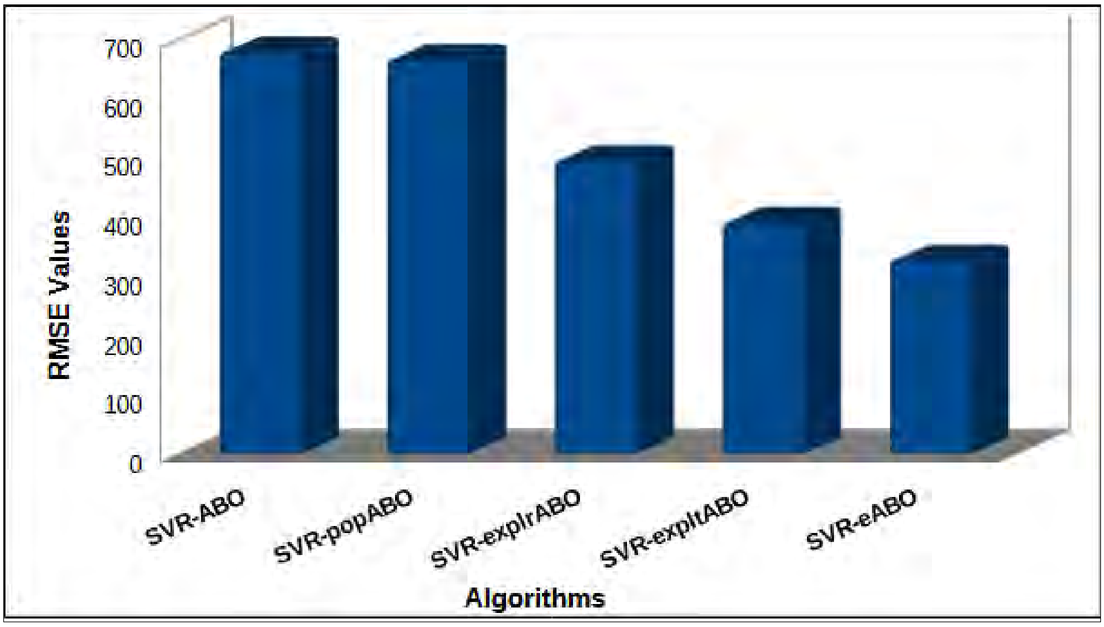

Figure 5.1: Comparison of RMSE (developed algorithms) on Household dataset

5.1.1.2 Mean Absolute Percentage Error (MAPE)

The MAPE measures the average percentage deviation between the predicted values and the actual values. It indicates the relative accuracy of the algorithms. SVR-*e*ABO achieved the lowest MAPE (1.4924), indicating the smallest average percentage deviation from the actual values. SVR-*explt*ABO (1.7844), SVR-*explr*ABO (2.2088), SVR-*pop*ABO (3.1949), and SVR-ABO (3.0544) follow with progressively higher MAPE values. Figure 5.2 presents visual comparison of MAPE values of the developed algorithms.

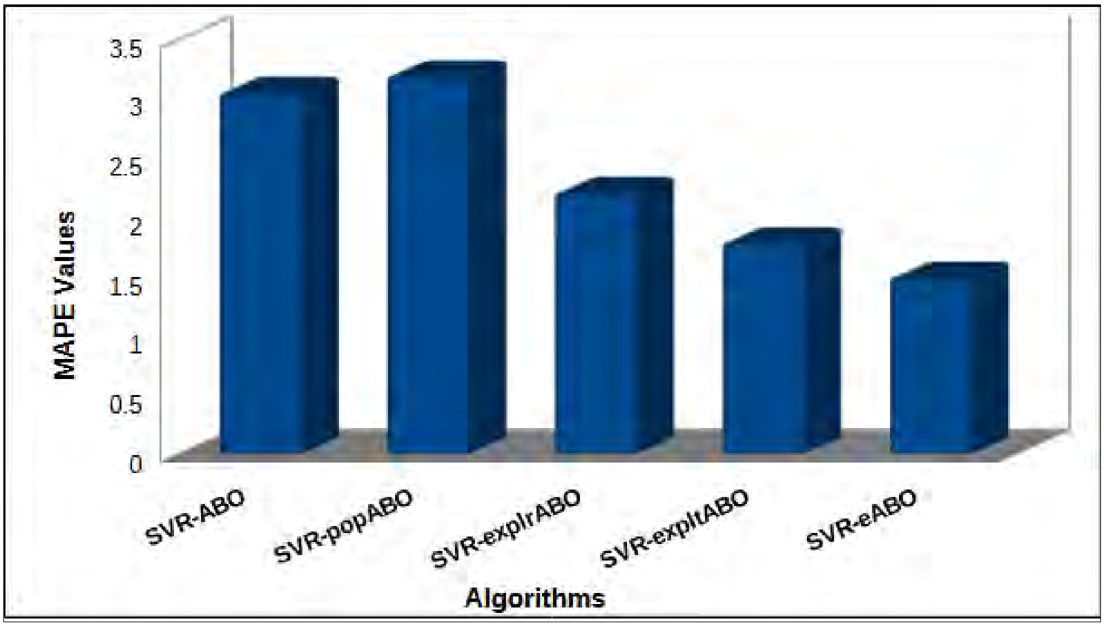

Figure 5.2: Comparison of MAPE (developed algorithms) on Household dataset

5.1.1.3 Mean Absolute Error (MAE)

The MAE measures the average deviation between the predicted values and the actual values. It provides a similar assessment to RMSE but without considering the squared values. SVR-eABO achieved the lowest MAE (239.2793), indicating the smallest average deviation from the actual values. SVR-*explt*ABO (273.9614), SVR-*explr*ABO (352.1861), SVR-*pop*ABO (521.2221), and SVR-ABO (514.9804) follow with progressively higher MAE values. Figure 5.3 shows a visual comparison of MAE values of the developed algorithms.

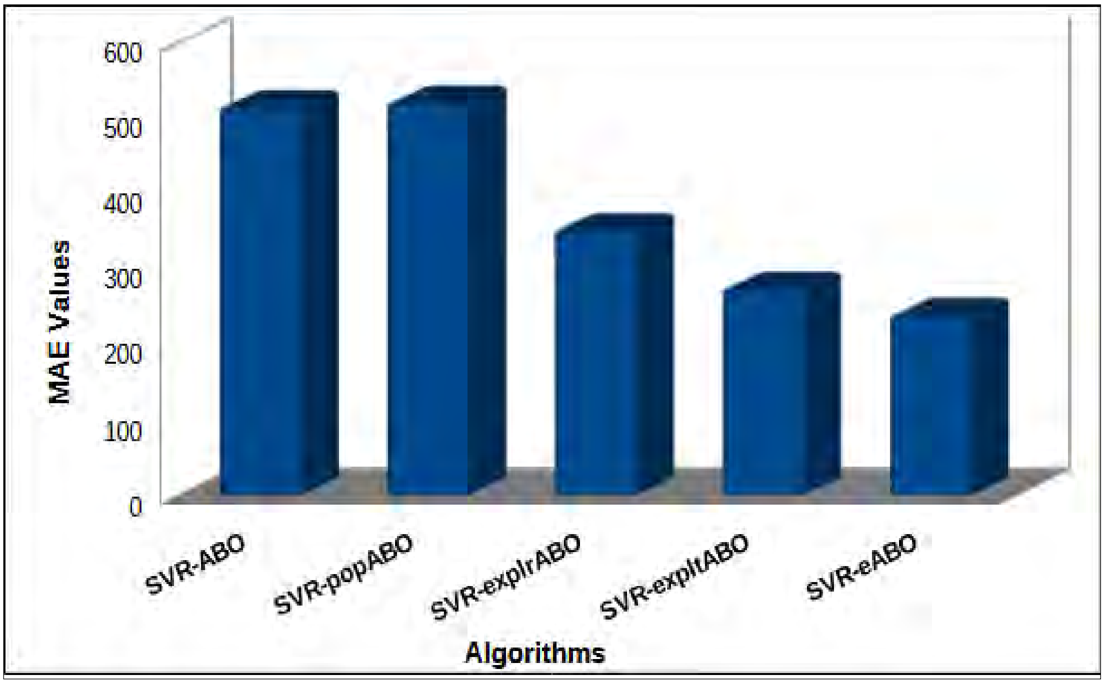

Figure 5.3: Comparison of MAE (developed algorithms) on Household dataset

5.1.1.4 Coefficient of Determination (R^2)

The R^2 value represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It quantifies the goodness-of-fit of the algorithms. SVR-eABO achieved the highest R^2 (0.9986), indicating the highest degree of predictability and goodness-of-fit. SVR-*explt*ABO (0.9980), SVR-*explr*ABO (0.9969), SVR-*pop*ABO (0.9943), and SVR-ABO (0.9941) follow with progressively lower R^2 values. Figure 5.4 demonstrate a visual comparison of R^2 values of the developed algorithms.

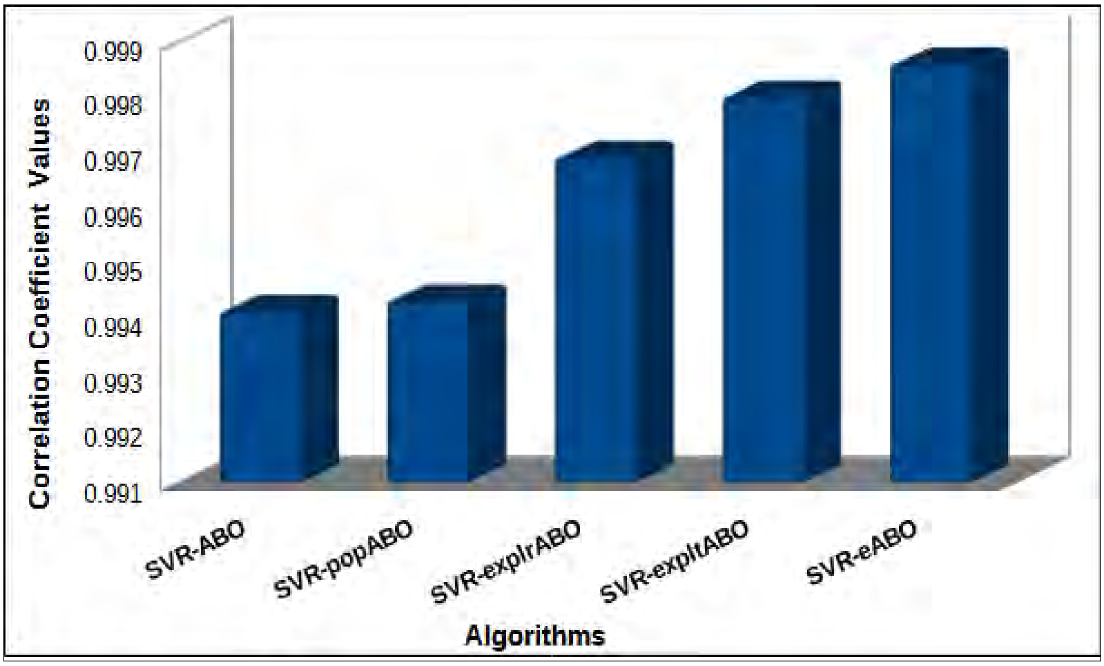

Figure 5.4: Comparison of R^2 (developed algorithms) on Household dataset

5.1.1.5 Percentage Accuracy (PA)

Percentage Accuracy represents the proportion of correctly predicted values. It provides an assessment of the overall accuracy of the algorithms. SVR-*e*ABO achieved the highest PA of 98.5076%, indicating the highest proportion of correct forecasting. SVR-*explt*ABO was able to record PA value of 98.2156%, SVR-*explr*ABO (97.7912%), SVR-*pop*ABO (96.8051%), and SVR-ABO (96.9456%) follow with progressively lower PA values. Figure 5.5 illustrates a visual comparison of PA values of the enhanced ABO algorithm against on Household dataset.

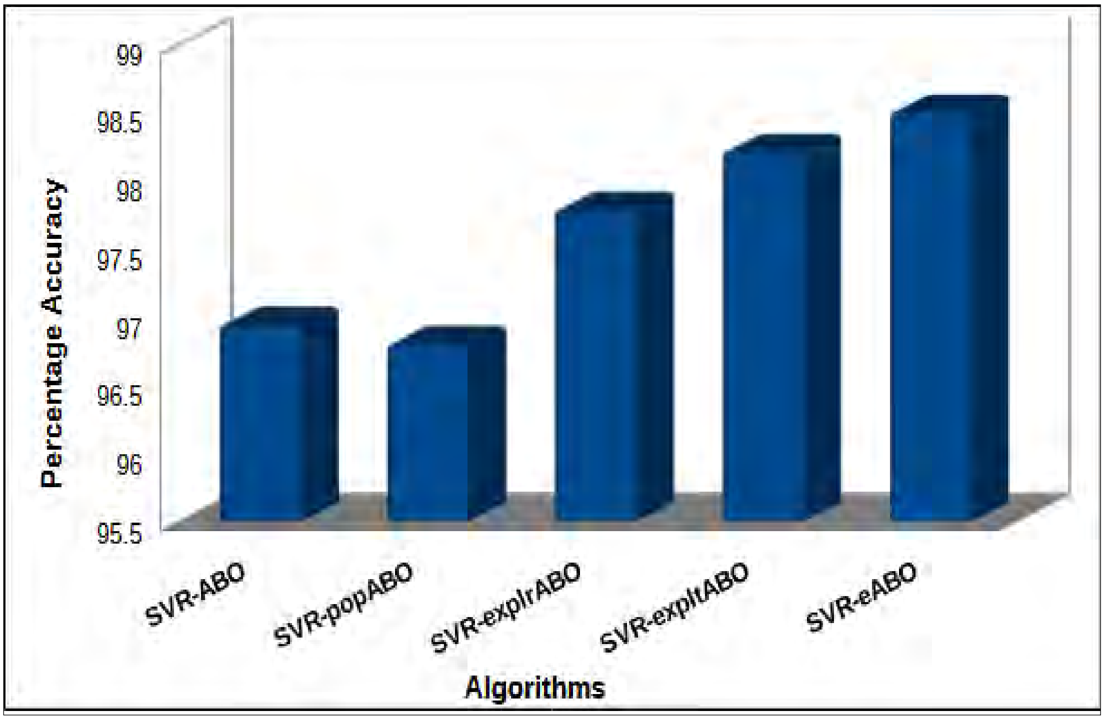

Figure 5.5: Comparison of PA (developed algorithms) on Household dataset

In summary, the enhanced algorithms (SVR-*pop*ABO, SVR-*explr*ABO, SVR-*explt*ABO, and SVR-*e*ABO) consistently outperformed the classical SVR-ABO algorithm based on RMSE, MAPE, MAE, R^2, and PA as evaluation metrics. SVR-*e*ABO achieved the best overall performance, with the lowest RMSE, MAPE, MAE, and the highest R^2 and PA values.

5.1.2 Comparison of SVR-*e*ABO Against Benchmarks on Household Dataset

In this section, the performance of the enhanced ABO algorithm as an SVR optimiser has been compared with selected benchmarks based on five evaluation metrics, the comparative performance of these benchmarks is as presented in table 5.2, while subsequent subheadings present detailed analysis of the result presented in the mentioned table.

Table 5.2

Comparative performance of eABO algorithm against Benchmarks

	SVR	**SVR-ABC**	**SVR-GA**	**SVR-PSO**	**SVR-CS**	**SVR-*e*ABO**
C	-	2717.5632	3537.7894	1045.0209	2917.6543	1949.6717
Epsilon	-	0.21342	0.2142	0.2017	0.1941	0.0979
Gamma	-	0.00634	0.0030	0.0029	0.0071	0.0018
RMSE	2222.0536	712.9428	773.2354	719.5689	732.8290	**327.4449**
MAPE	3.8343	3.2897	3.7564	3.4704	3.5261	**1.4924**
MAE	1008.7593	511.3731	591.0755	544.1635	537.1287	**239.2793**
R^2	0.9366	0.9935	0.9923	0.9934	0.99041	**0.9986**
PA (%)	96.1657	96.7103	96.2436	96.5296	96.4739	**98.5076**
CPU Time	**0.0287**	15.2454	14.6549	18.0093	16.9270	24.4815

5.1.2.1 Root Mean Square Error (RMSE)

Lower RMSE values indicate better predictive accuracy, with less deviation between the predicted and actual values. SVR-eABO achieves the lowest RMSE value among all the algorithms, indicating superior predictive accuracy. SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS also show lower RMSE values compared to SVR, suggesting improved accuracy in predicting the target variable. Figure 5.6 illustrates a visual comparison of RMSE values of the enhanced ABO algorithm against benchmarked algorithms.

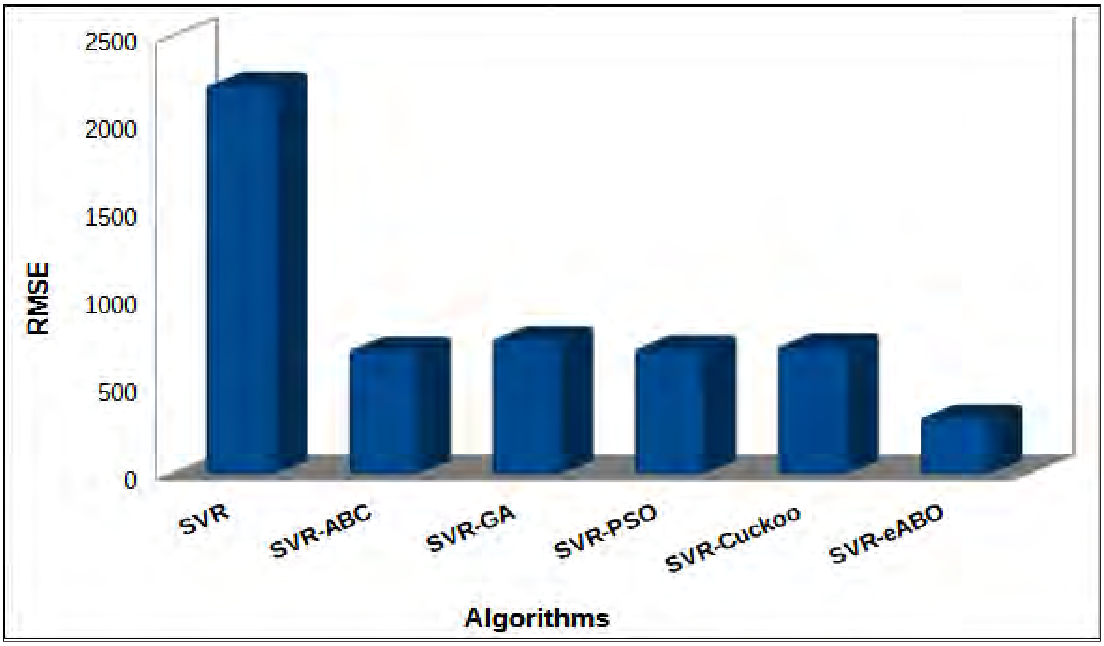

Figure 5.6: Comparison of RMSE (against Benchmarks) on Household dataset

5.1.2.2 Mean Absolute Percentage Error (MAPE)

MAPE measures the average percentage difference between the predicted and actual values. Lower MAPE values indicate better predictive accuracy, with less relative error in forecasting. SVR-eABO achieves the lowest MAPE value among all the algorithms, indicating superior accuracy in predicting the target variable. SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS also show lower MAPE values compared to SVR, suggesting improved accuracy in predicting relative errors. Figure 5.7 shows a visual comparison of MAPE values of the enhanced ABO algorithm against benchmarked algorithms.

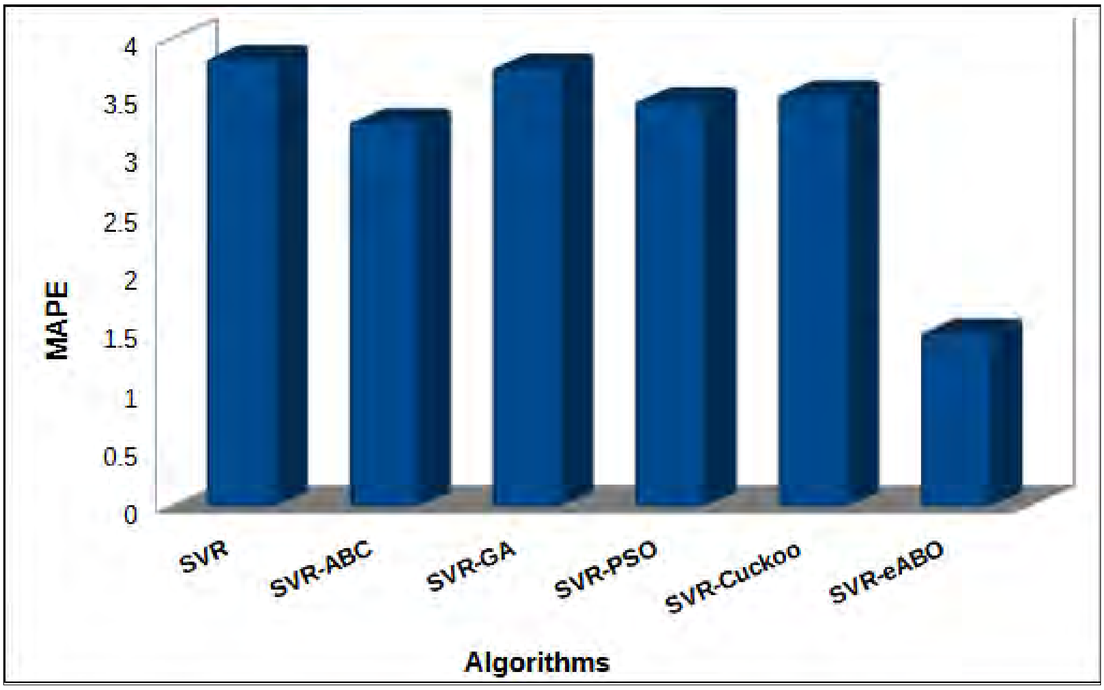

Figure 5.7: Comparison of MAPE (against Benchmarks) on Household dataset

5.1.2.3 Mean Absolute Error (MAE)

MAE measures the average absolute difference between the predicted and actual values. Lower MAE values indicate better predictive accuracy, with less absolute error in forecasting. SVR-*e*ABO achieves the lowest MAE value among all the algorithms, indicating superior accuracy in predicting the target variable. SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS also show lower MAE values compared to SVR, suggesting improved accuracy in predicting absolute errors. Figure 5.8 depicts a visual comparison of MAE values of the enhanced ABO algorithm against benchmarked algorithms.

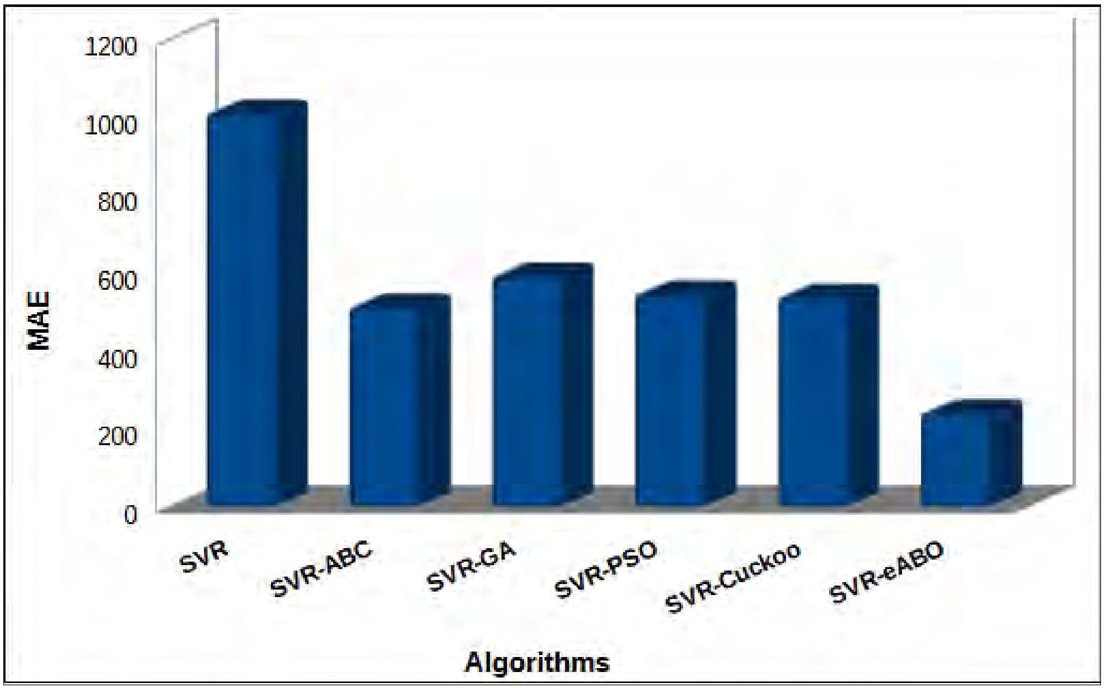

Figure 5.8: Comparison of MAE (against Benchmarks) on Household dataset

5.1.2.4 Coefficient of Determination (R^2)

Coefficient of Determination (R^2) measures the proportion of variance in the target variable that can be explained by the model. Higher R^2 values indicate a better fit of the model to the data. SVR-*e*ABO achieves the highest R^2 value among all the algorithms, indicating the best overall model fit. SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS also show higher R^2 values compared to SVR, suggesting improved model fit. Figure 5.9 presents a visual comparison of Coefficient of Determination values of the enhanced ABO algorithm against benchmarked algorithms.

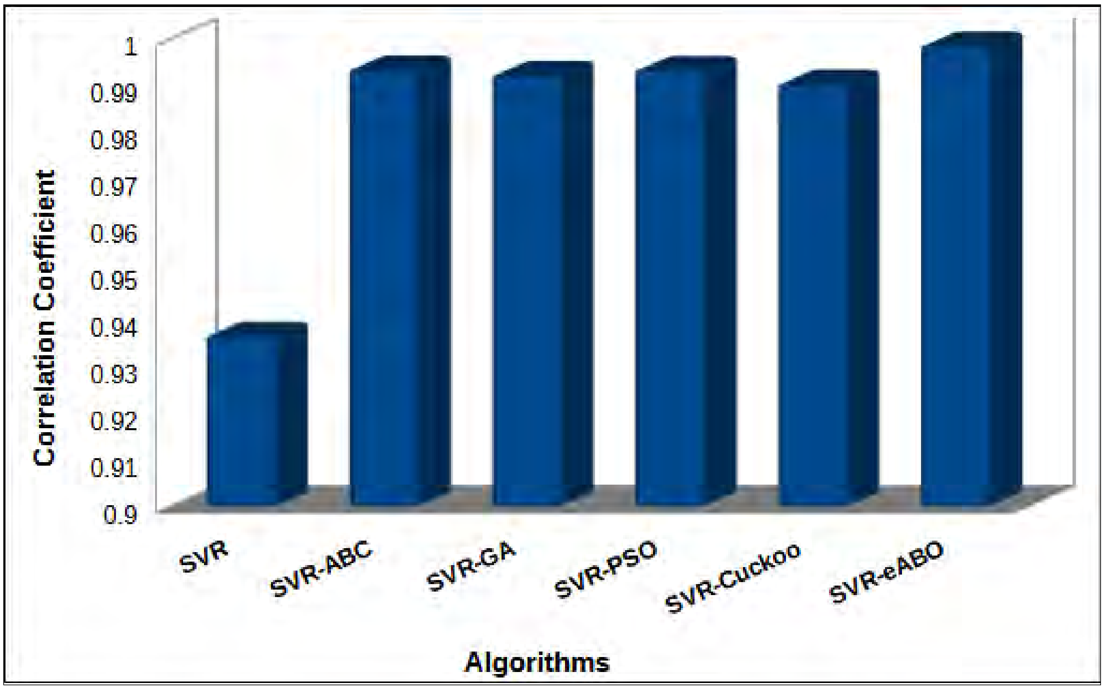

Figure 5.9: Comparison of R^2 (against Benchmarks) on Household dataset

5.1.2.5 Percentage Accuracy (PA)

Percentage Accuracy (PA) represents the percentage of accurate forecasting made by the model. Higher PA % values indicate better precision in predicting the correct outcomes. SVR-*e*ABO achieves the highest PA % value among all the algorithms, indicating superior precision. SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS also show higher PA values compared to SVR, suggesting improved precision. Figure 5.10 depicts a visual comparison of MAE values of the enhanced ABO algorithm against benchmarked algorithms.

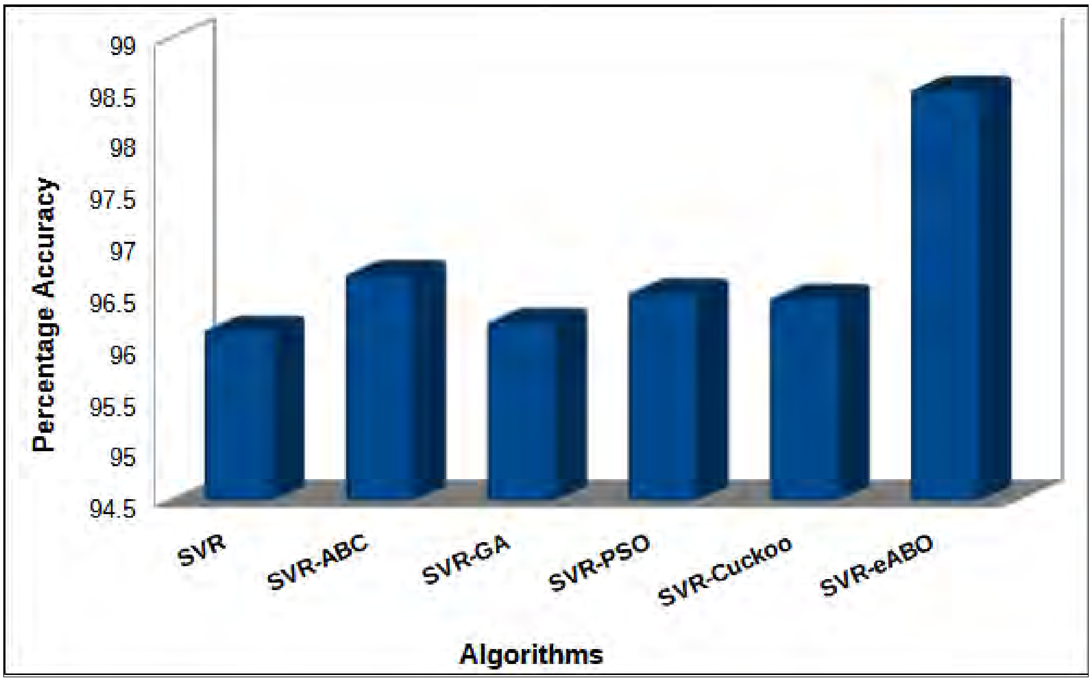

Figure 5.10: Comparison of PA (against Benchmarks) on Household dataset

SVR-*e*ABO consistently outperforms the other algorithms in terms of RMSE, MAPE, MAE, R^2, and PA. The algorithm demonstrates the best overall predictive accuracy, model fit, and precision among the tested algorithms.

5.1.2.6 CPU Execution Time

The time taken by each algorithm has also been evaluated. The results shows that classical SVR algorithm was able to finish execution within the least amount of time. It is followed by SVR-GA and SVR-ABC with CPU time of 14.6549 and 15.2454 seconds respectively. While Figure 5.11 shows a visual comparison of MAE values of the enhanced ABO algorithm against benchmarked algorithms.

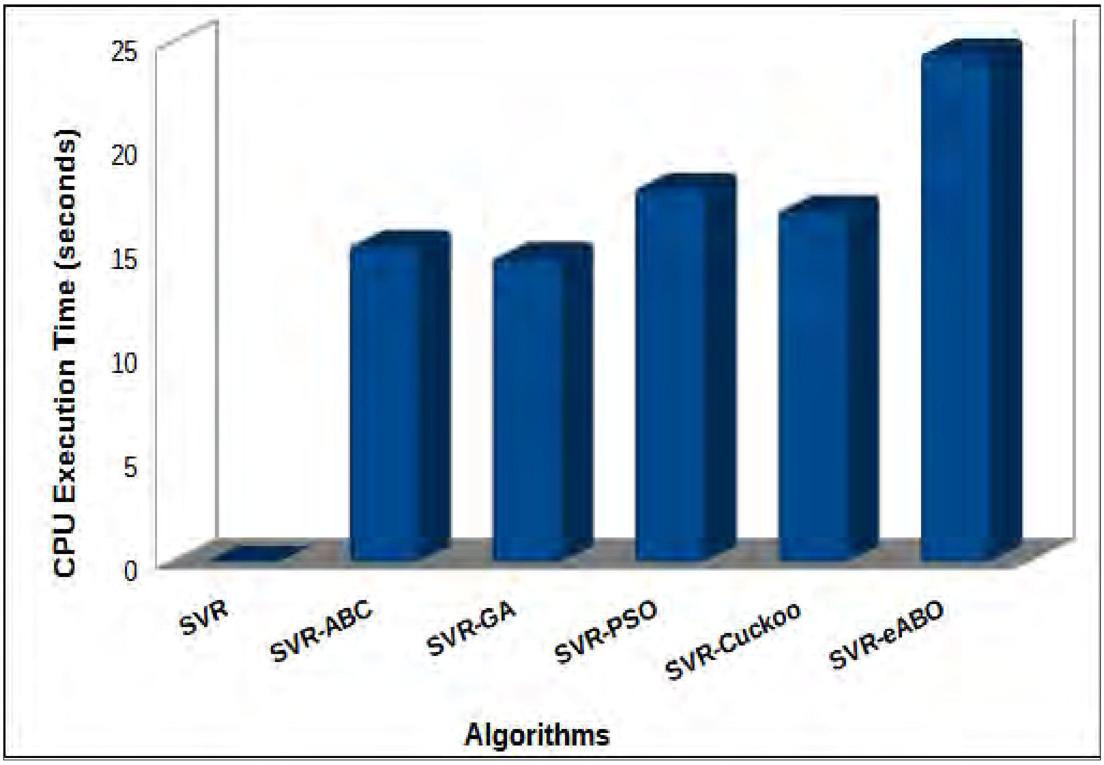

Figure 5.11: Comparison of CPU Time (against Benchmarks) on Household dataset

5.2 Turkey dataset

In this section the performance of all enhanced algorithms was compared on Turkey dataset likewise, the performance of the final algorithm (SVR-*e*ABO) has been compared with the selected benchmarks on the same dataset. Below are the performances based on enhancement at each stage and in comparison, to the benchmarks.

5.2.1 Comparison Between Algorithms on Turkey dataset

This section presents the results obtained from evaluating the five algorithms developed in this study on Turkey dataset. The obtained results of the evaluations are presented below in table 5.3, while the interpretation and analysis of the result followed in subsequent sub-sections.

Table 5.3

Comparative performance of algorithms on Turkey Dataset

	SVR-ABO	**SVR-*pop*ABO**	**SVR-*explr*ABO**	**SVR-*explt*ABO**	**SVR-*e*ABO**
C	2249.2718	1440.0861	1468.0306	4.10072	2161.9007
Epsilon	0.091542	0.3000	0.04879	0.0706	0.0915
Gamma	0.0989	0.0089	0.09902	0.00031	0.0097
RMSE	531.6825	473.3348	436.3436	367.2885	**298.6726**
MAPE	3.0895	2.9468	2.4841	2.1836	**1.8522**
MAE	395.0339	376.7470	317.2447	288.4540	**238.1015**
R^2	0.7908	0.8342	0.8591	0.90017	**0.93398**
PA (%)	96.9106	97.0532	97.5159	97.8164	**98.1478**
CPU Time	**41.8235**	45.4421	52.56740	84.1104	61.8235

5.2.1.1 Root Mean Square Error (RMSE)

Lower RMSE values indicate better predictive accuracy, with less deviation between the predicted and actual values. SVR-*e*ABO consistently achieves the lowest RMSE value among all the variants, suggesting it has the best overall predictive accuracy. SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO also show lower RMSE values compared to SVR, indicating improved accuracy in predicting the target variable. SVR-ABO exhibits a slightly higher RMSE value, indicating a potential limitation in terms of predictive accuracy. Figure 5.12 depict a visual comparison of RMSE values of the developed algorithms.

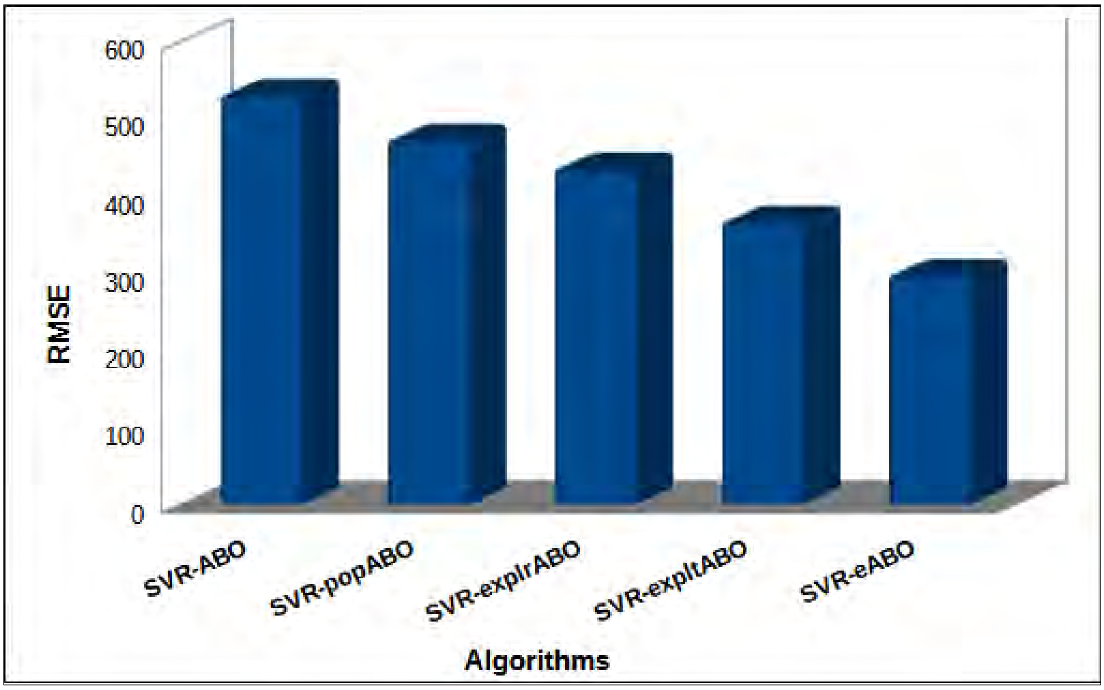

Figure 5.12: Comparison of RMSE (developed algorithms) on Turkey dataset

5.2.1.2 Mean Absolute Percentage Error (MAPE)

MAPE measures the average percentage difference between the predicted and actual values. Lower MAPE values indicate better predictive accuracy, with less relative error in forecasting. SVR-*e*ABO consistently achieves the lowest MAPE value among all the variants, indicating superior accuracy in predicting the target variable. SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO also show lower MAPE values compared to SVR, suggesting improved accuracy in predicting relative errors. SVR-ABO displays a slightly higher MAPE value, indicating a potential limitation in predicting relative errors. . Figure 5.13 depict a visual comparison of MAPE values of the developed algorithms.

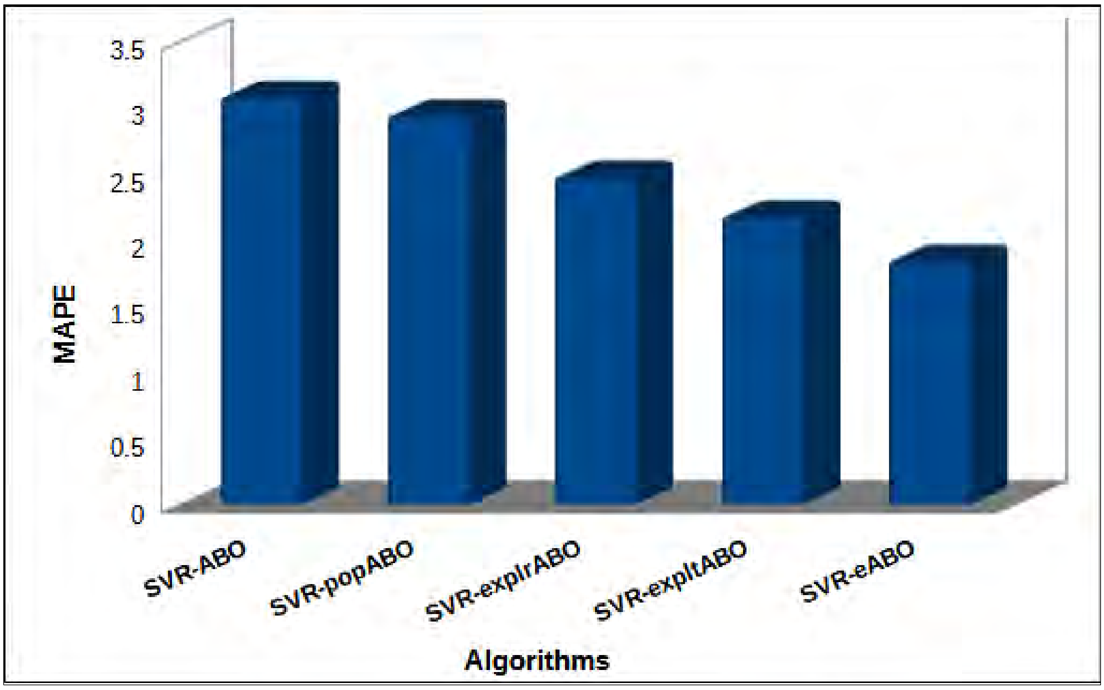

Figure 5.13: Comparison of MAPE (developed algorithms) on Turkey dataset

5.2.1.3 Mean Absolute Error (MAE)

MAE measures the average absolute difference between the predicted and actual values. Lower MAE values indicate better predictive accuracy, with less absolute error in forecasting. SVR-*e*ABO consistently achieves the lowest MAE value among all the variants, indicating superior accuracy in predicting the target variable. SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO also show lower MAE values compared to SVR, suggesting improved accuracy in predicting absolute errors. SVR-ABO exhibits a slightly higher MAE value than SVR, indicating a potential limitation in predicting absolute errors. Figure 5.14 depict a visual comparison of MAE values of the developed algorithms.

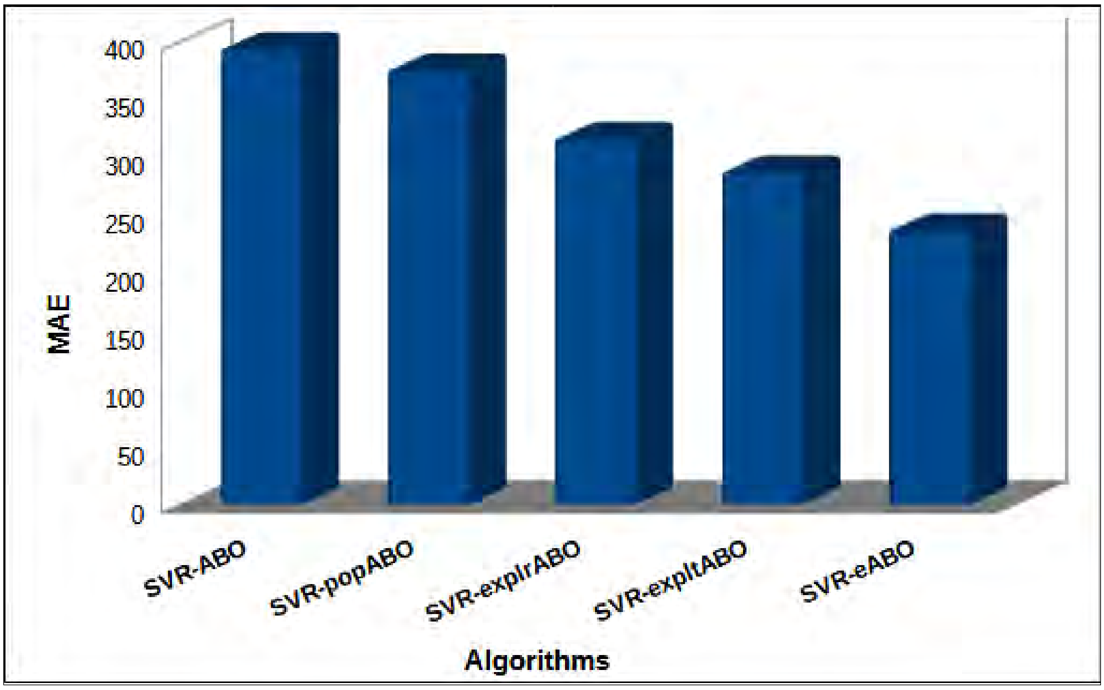

Figure 5.14: Comparison of MAE (developed algorithms) on Turkey dataset

5.2.1.4 Coefficient of Determination (R^2)

R^2 measures the proportion of variance in the target variable that can be explained by the model. Higher R^2 values indicate a better fit of the model to the data. SVR-eABO consistently achieves the highest R^2 value among all the variants, indicating the best overall model fit. SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO also show higher R^2 values compared to SVR, suggesting improved model fit. SVR-ABO displays a slightly lower R^2 value than SVR, indicating a potential limitation in terms of model fit. Figure 5.15 depict a visual comparison of Coefficient of Determination (R^2) values of the developed algorithms.

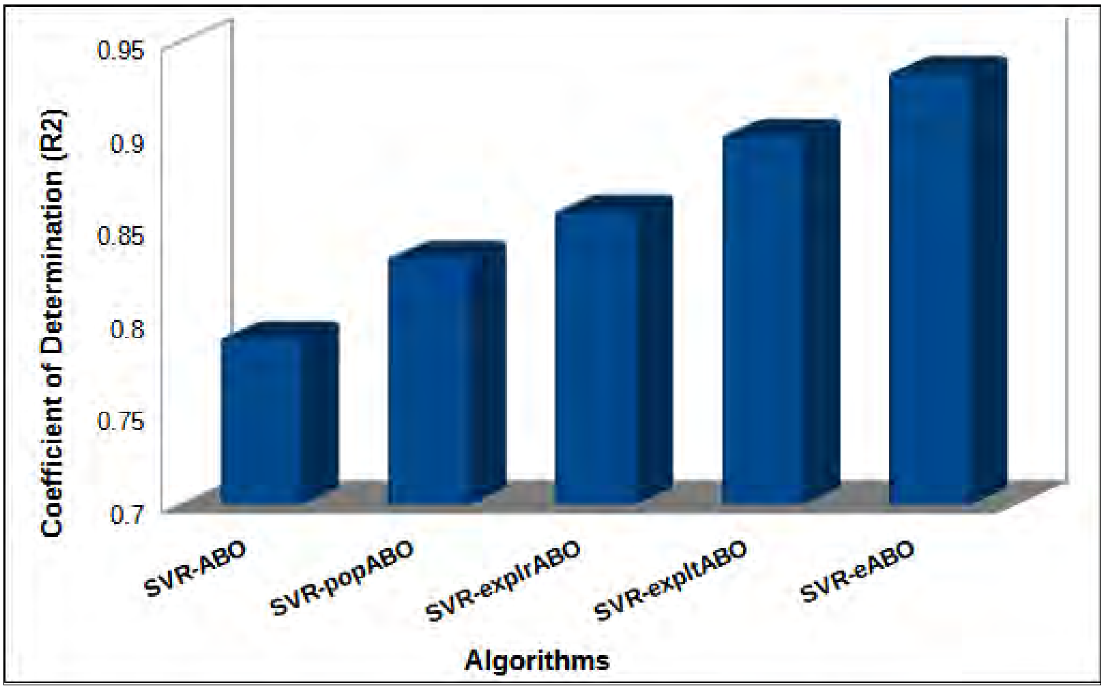

Figure 5.15: Comparison of R^2 (developed algorithms) on Turkey dataset

5.2.1.5 Percentage Accuracy (PA)

Percentage Accuracy represents the percentage of accurate forecasting made by the model. Higher PA values indicate better precision in predicting the correct outcomes. SVR-*e*ABO consistently achieves the highest PA value among all the variants, indicating superior precision. SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO also show higher PA % values compared to SVR, suggesting improved precision. SVR-ABO exhibits a slightly lower PA value than SVR, indicating a potential limitation in terms of precision. . Figure 5.16 depict a visual comparison of Percentage Accuracy of the developed algorithms.

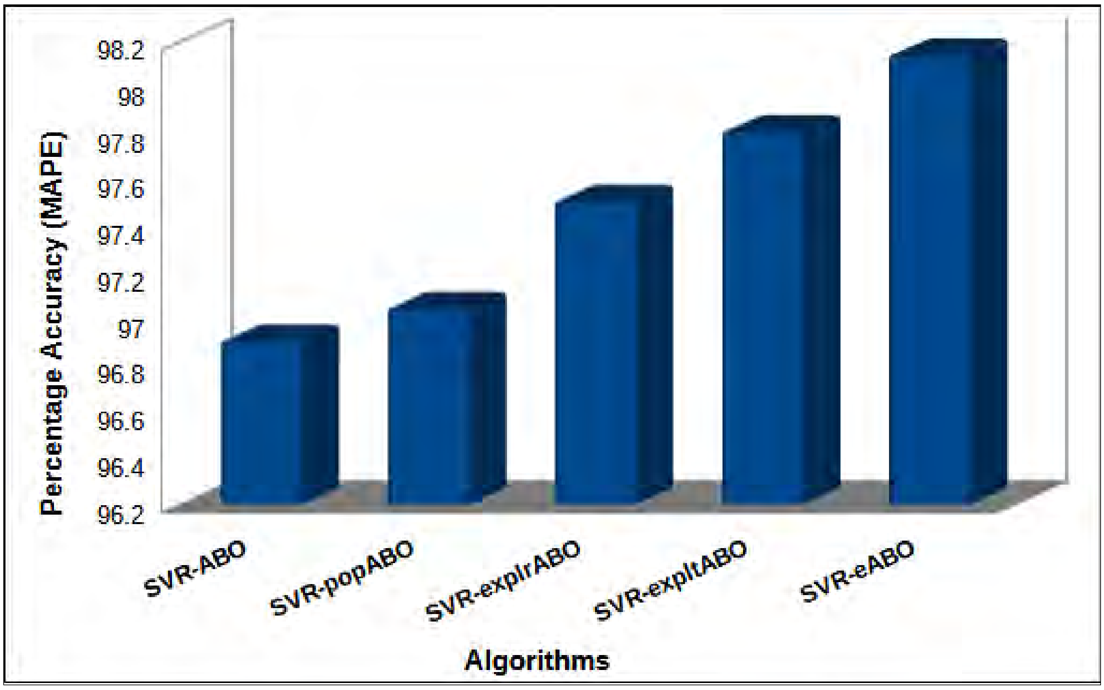

Figure 5.16: Comparison of PA (developed algorithms) on Turkey dataset

In summary, SVR-eABO consistently outperforms the other variants in terms of RMSE, MAPE, MAE, R^2, and PA. It demonstrates the best overall predictive accuracy, model fit, and precision among the tested variants. SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO also show improvements in these metrics compared to the conventional SVR model, although to a lesser extent than SVR-*e*ABO. SVR-ABO, on the other hand, exhibits performance comparable to or slightly worse than SVR.

5.2.2 Comparison of SVR-*e*ABO Against Benchmarks on Turkey Dataset

In this section, the performance of the enhanced ABO algorithm as an SVR optimiser has been compared with selected benchmarks based on five evaluation metrics, the comparative performance is as presented in table 5.4 and detailed analysis is presented in subsequent subheadings.

Table 5.4

Comparison of algorithm with benchmarks on Turkey Dataset

	SVR	SVR-ABC	SVRGA	SVR-PSO	SVR-CS	SVR-*e*ABO
C	-	340.9366	303.7596	343.2435	409.8105	2161.9007
Epsilon	-	0.1029	0.1029	0.0977	0.7901	0.0915
Gamma	-	0.0996	0.0994	0.0948	0.8941	0.0097
RMSE	549.2810	548.6059	547.9467	532.4462	337.3901	**298.6726**
MAPE	3.2048	3.1903	3.1866	3.0970	2.9386	**1.8522**
MAE	409.7200	407.8633	407.3945	395.9976	287.9617	**238.1015**
R^2	0.7767	0.7773	0.7778	0.7902	0.9251	**0.93398**
PA (%)	96.7952	96.8097	96.8134	96.9030	97.0614	**98.1478**
CPU Time	**0.0014**	46.6948	47.4956	42.1758	45.9268	61.8235

5.2.2.1 Root Mean Square Error (RMSE)

SVR-*e*ABO achieves the lowest RMSE value (298.6726), indicating superior accuracy in predicting the target variable compared to the other algorithms. A lower RMSE signifies that the predicted values are closer to the actual values, reflecting better overall model performance. SVR-PSO (532.4462) and SVR-GA (547.9467) also demonstrate relatively lower RMSE values, suggesting good predictive accuracy. SVR-ABC (548.6059), SVR-CS (337.3901), and SVR (549.2810) have slightly higher RMSE values, indicating larger forecasting errors. Figure 5.17 shows a visual comparison of all benchmarked algorithms' performances based on RMSE metric on Turkey dataset.

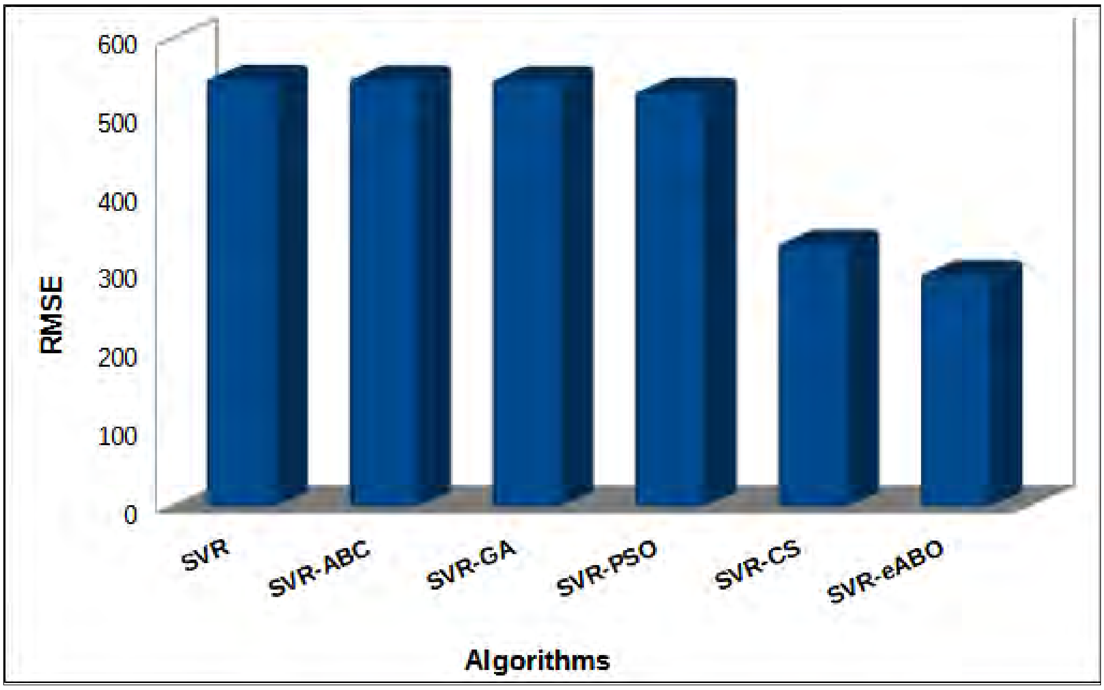

Figure 5.17: Comparison of RMSE (against Benchmarks) on Turkey dataset

5.2.2.2 Mean Absolute Percentage Error (MAPE)

SVR-*e*ABO achieves the lowest MAPE value (1.8522), indicating the smallest average percentage difference between the predicted and actual values. A lower MAPE signifies better accuracy and a better fit of the model. SVR-PSO (3.0970), SVR-GA (3.1866), and SVR-ABC (3.1903) also exhibit relatively low MAPE values, suggesting good predictive accuracy. SVR-CS (2.9386) and SVR (3.2048) have slightly higher MAPE values, indicating a slightly larger average percentage difference between the predicted and actual values. Figure 5.18 shows a visual comparison of values recorded by the SVR-eABO and the benchmarked algorithms based on MAPE metric.

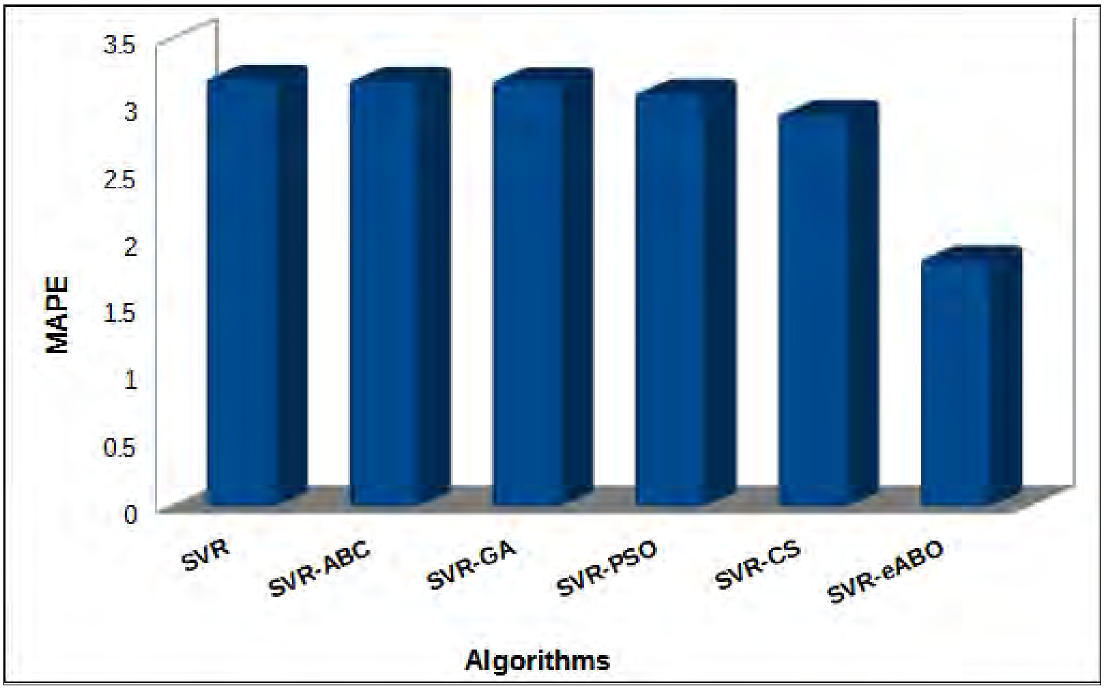

Figure 5.18: Comparison of MAPE (against Benchmarks) on Turkey dataset

5.2.2.3 Mean Absolute Error (MAE)

SVR-*e*ABO achieves the lowest MAE value (238.1015), indicating the smallest average deviation between the predicted and actual values. A lower MAE suggests better accuracy and a better fit of the model. SVR-PSO (395.9976) and SVR-GA (407.3945) also demonstrate relatively low MAE values, suggesting good predictive accuracy. SVR-ABC (407.8633), SVR-CS (287.9617), and SVR (409.7200) have slightly higher MAE values, indicating a slightly larger average deviation between the predicted and actual values. Comparative analysis is as presented in figure 5.19.

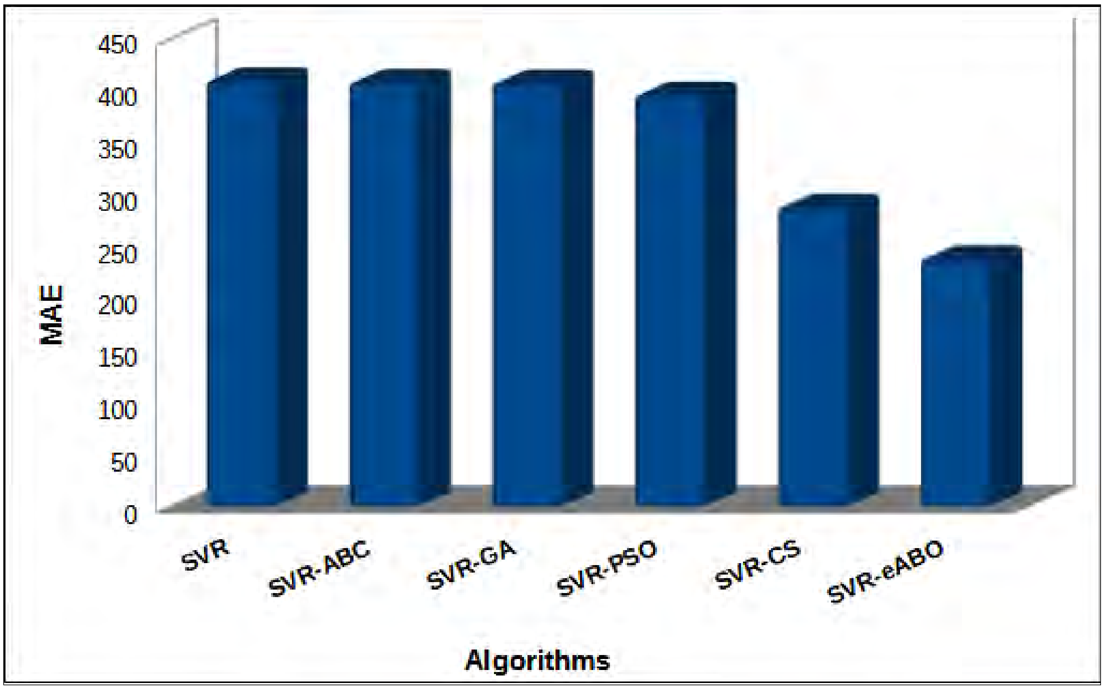

Figure 5.19: Comparison of MAE (against Benchmarks) on Turkey dataset

5.2.2.4 Coefficient of Determination (R^2)

SVR-*e*ABO achieves the highest R^2 value (0.93398), indicating the best goodness-of-fit compared to the other algorithms. A higher R^2 value signifies that a larger proportion of the variance in the dependent variable is explained by the independent variables. SVR (0.7767), SVR-ABC (0.7773), and SVR-GA (0.7778) also demonstrate relatively high R^2 values, suggesting good explanatory power. SVR-PSO (0.7902) and SVR-CS (0.9251) have slightly lower R^2 values, indicating relatively less variance explained by the independent variables as presented in figure 5.20.

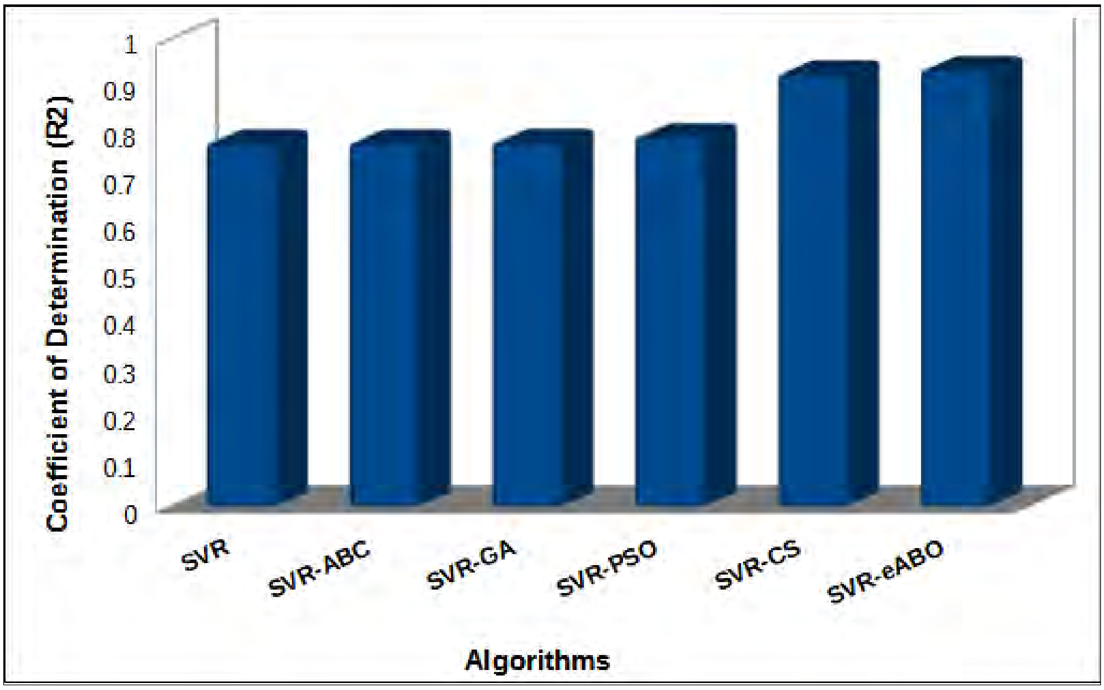

Figure 5.20: Comparison of R^2 (against Benchmarks) on Turkey dataset

5.2.2.5 Percentage Accuracy (PA)

SVR-*e*ABO achieves the highest PA value (98.1478%), indicating the highest accuracy in predicting instances. A higher PA value suggests a higher proportion of correctly predicted instances. SVR-CS (97.0614%) and SVR-ABC (96.8097%) also demonstrate relatively high PA values, indicating good predictive accuracy. SVR-GA (96.8134%), SVR-PSO (96.9030%), and SVR (96.7952%) have slightly lower PA values, suggesting a slightly lower proportion of correctly predicted instances. Figure 5.21 presents the visual depiction of the algorithms' performances based on PA values.

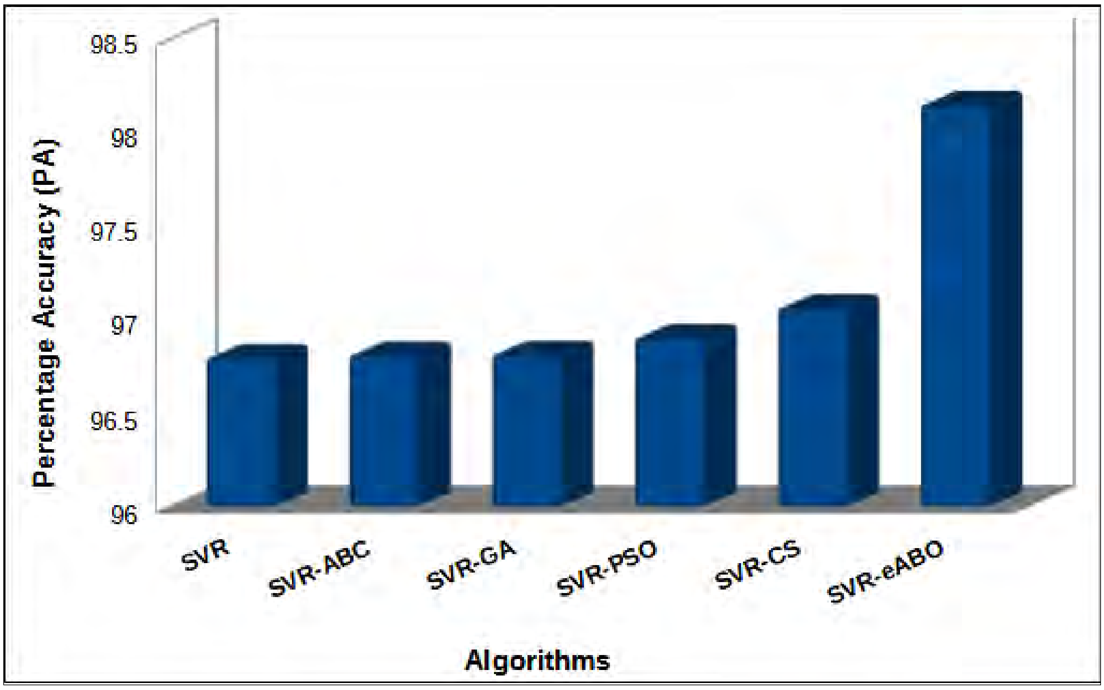

Figure 5.21: Comparison of PA (against Benchmarks) on Turkey dataset

5.2.2.6 CPU Execution Time

SVR-PSO has the lowest duration (42.1758) among the algorithms apart from the classical SVR algorithm that records overall lowest, indicating the shortest execution time among the algorithms as presented in figure 5.22. A lower duration suggests faster processing speed. SVR-GA (47.4956) also demonstrates relatively low duration, indicating fast execution. SVR-CS (45.9268) and SVR-ABC (46.6948) have slightly higher durations, suggesting slightly longer execution times. SVR-*e*ABO has the highest duration (61.8235), indicating the longest execution time.

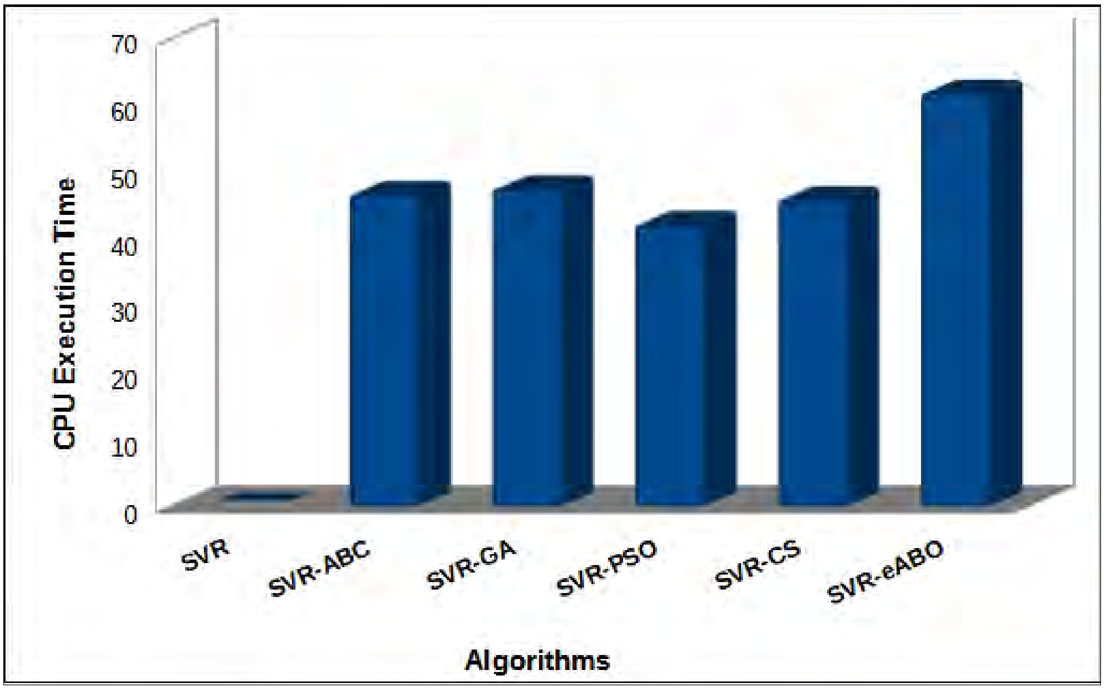

Figure 5.22: Comparison of CPU Time (against Benchmarks) on Turkey dataset

In summary, SVR-*e*ABO consistently demonstrates superior performance across multiple metrics, including the lowest RMSE, MAPE, MAE values, highest R^2 value, highest PA, and relatively longer duration. This algorithm shows excellent accuracy, good model fit, and a high proportion of correctly predicted instances but requires a longer execution time. SVR-PSO and SVR-GA also exhibit competitive performance, with relatively low RMSE, MAPE, MAE values, high R^2 values, and relatively shorter duration. These algorithms provide good accuracy, decent model fit, and a high proportion of correctly predicted instances while being computationally efficient. SVR-ABC, SVR-CS, and SVR perform relatively well in terms of RMSE, MAPE, and MAE values, with slightly higher values compared to SVR-eABO, SVR-PSO, and SVR-GA. However, they have lower R^2 values and slightly lower PA. SVR-ABC and SVR-CS have relatively shorter execution times, while SVR has a shortest execution time.

Overall, SVR-*e*ABO stands out as the best-performing algorithm in terms of accuracy and model fit, but it comes at the cost of longer execution time. SVR-PSO and SVR-

GA provide a good balance between accuracy and execution time. SVR-ABC, SVR-CS, and SVR also offer decent performance, but they may have slightly lower accuracy and model fit compared to the top-performing algorithms.

5.3 Appliances dataset

In this section, a comparative analysis was conducted to assess the performance of various enhanced algorithms on the Appliances dataset. Additionally, the performance of the final algorithm (SVR-*e*ABO) was evaluated in comparison to the developed hybrid algorithms at various stages mentioned using the same dataset as presented in table 5.5. The subsequent results outline the performance enhancements achieved at each stage and provide a comparative assessment against the final SVR-*e*ABO.

5.3.1 Comparison Between Developed Algorithms on Appliances dataset

This section presents the results obtained from evaluating the five algorithms developed in this study on Appliances dataset. The obtained results of the evaluations are presented below in table 5.5, while the interpretation and analysis of the result followed in subsequent sub-sections.

Table 5.5

Comparison of developed Algorithms on Appliances dataset

	SVR-ABO	**SVR - *pop*ABO**	**SVR-*explr*ABO**	**SVR-*explt*ABO**	**SVR-*e*ABO**
C	11.3242	0.44910	0.2349	1.7912	3.0768
Epsilon	0.0934	0.02340	0.0206	0.0619	0.0534
Gamma	0.0030	0.02713	0.02321	0.0027	0.0009
RMSE	453.1873	430.0246	**425.2804**	452.6258	434.9740
MAPE	9.8529	9.6602	9.0144	**8.6573**	8.8263
MAE	0.3370	0.3473	0.3229	0.3166	**0.3112**

R^2	0.1311	0.2176	**0.2350**	0.1327	0.1994
PA (%)	90.1471	90.3398	90.9856	**91.3427**	91.1737
CPU Time	358.9556	404.8427	**331.3231**	421.3143	435.0065

5.3.1.1 Root Mean Square Error (RMSE)

RMSE as a metric that measures the average magnitude of forecasting errors has been employed to measure the performance of the developed algorithms. Lower RMSE values, such as those achieved by SVR-expltABO (0.4252) and SVR-*e*ABO (0.4349), indicate better accuracy and smaller forecasting errors. This suggests that the predicted values are closer to the actual values. In contrast, higher RMSE values, as seen in SVR-popABO (0.43), SVR-ABO (0.4531), and SVR (0.4559), indicate larger forecasting errors. This may imply that the models have more difficulty to accurately predict the target variable. Figure 5.23 depict a visual comparison of RMSE values of the developed algorithms.

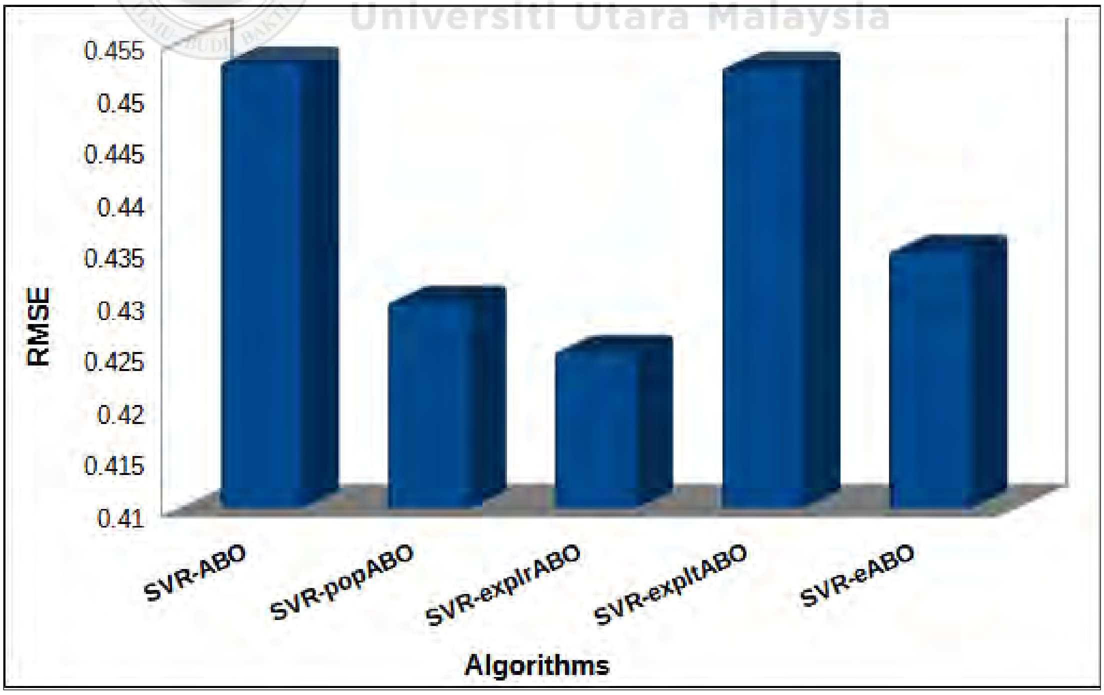

Figure 5.23: Comparison of RMSE on Appliances dataset

5.3.1.2 Mean Absolute Percentage Error (MAPE)

MAPE measures the average percentage difference between predicted and actual values. Lower MAPE values, such as those achieved by SVR-*explt*ABO (8.6573) and SVR-eABO (8.8263), indicate smaller average percentage differences and better accuracy. Higher MAPE values, seen in SVR-popABO (9.6602), SVR-ABO (9.8529), and SVR (9.7568), suggest larger average percentage differences, indicating a higher degree of deviation between the predicted and actual values. Figure 5.24 depict a visual comparison of MAPE values of the developed algorithms.

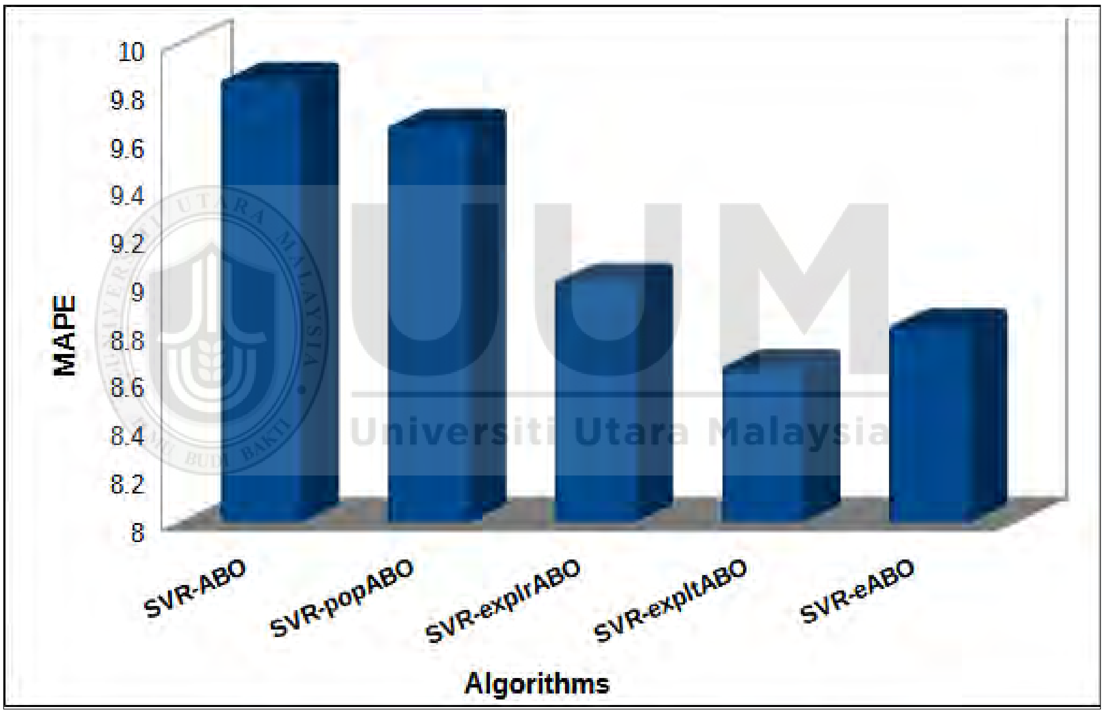

Figure 5.24: Comparison of MAPE on Appliances dataset

5.3.1.3 Mean Absolute Error (MAE)

MAE measures the average magnitude of forecasting errors without considering their direction. A lower MAE value, as achieved by SVR-*e*ABO (0.3112), indicates smaller average deviations between the predicted and actual values. Slightly higher MAE values in SVR-ABO (0.3370), SVR-*pop*ABO (0.3473), and SVR (0.3742) suggest slightly larger average deviations. Figure 5.25 depict a visual comparison of MAE values of the developed algorithms.

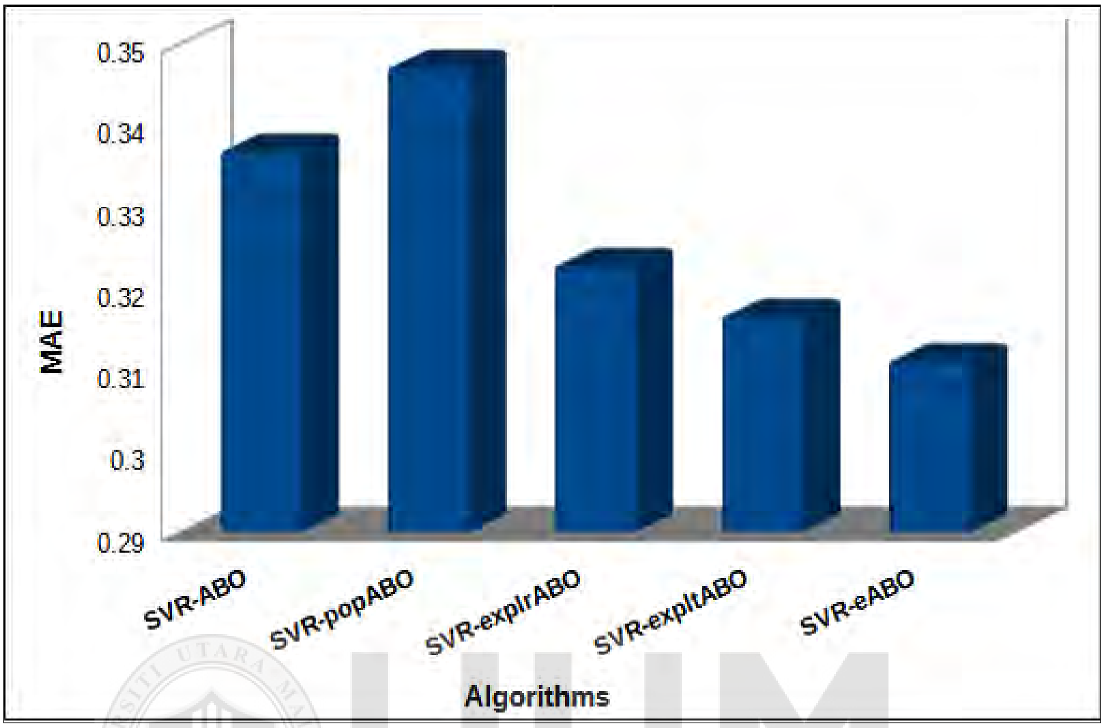

Figure 5.25: Comparison of MAE on Appliances dataset

5.3.1.4 Coefficient of Determination (R^2)

The coefficient of determination (R^2) is utilized to assess the degree to which a model fits the data. It quantifies the proportion of variance in the dependent variable that can be explained by the independent variables. A higher R^2 value, exemplified by SVR-*e*ABO (0.8994), signifies a superior fit and a greater proportion of explained variance. Conversely, lower R^2 values observed in SVR-popABO (0.8176), SVR-ABO (0.8311), and SVR (0.8203) indicate a relatively less variance explained by the independent variables and a lower level of goodness-of-fit. Figure 5.26 presents a visual comparison of the R^2 values achieved by the developed algorithms.

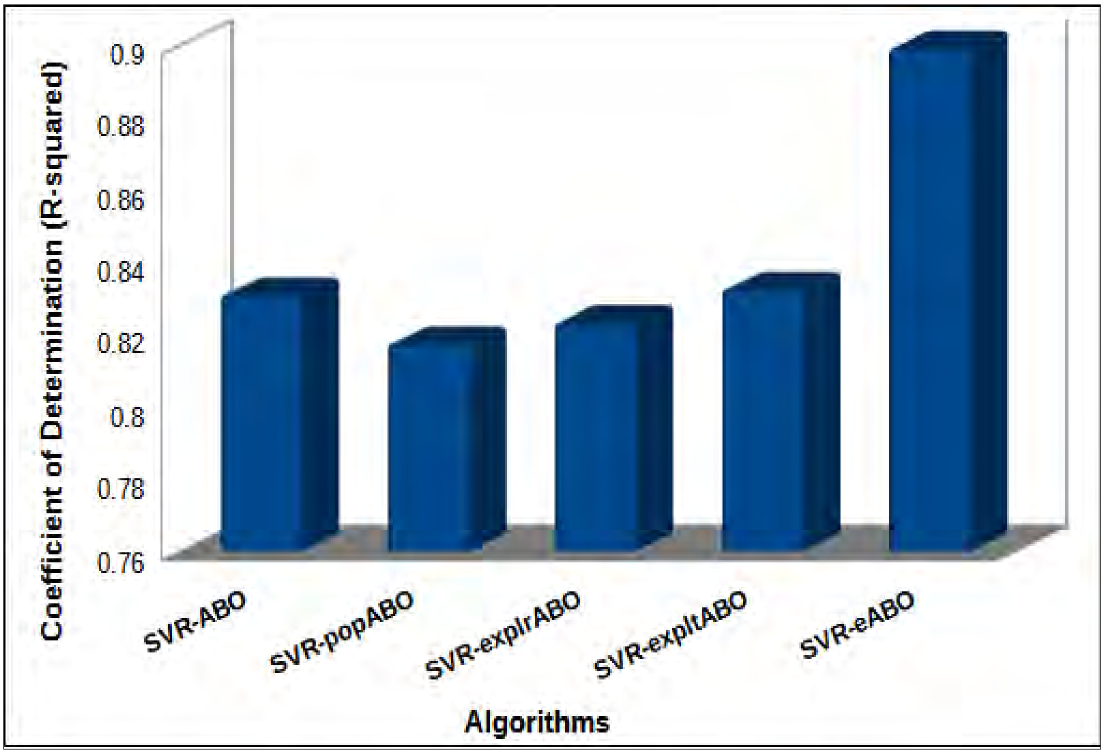

Figure 5.26: Comparison of R^2 on Appliances dataset

5.3.1.5 Percentage Accuracy (PA)

Percentage Accuracy (PA) measures the percentage of correctly predicted instances. A higher PA value, such as that achieved by SVR-*e*ABO (91.1737%), indicates a higher accuracy in predicting instances. Slightly lower PA values, observed in SVR-*pop*ABO (90.3398%), SVR-ABO (90.1471%), and SVR (90.2432%), suggest a slightly lower proportion of correctly predicted instances. Figure 5.27 depict a visual comparison of PA values of the developed algorithms.

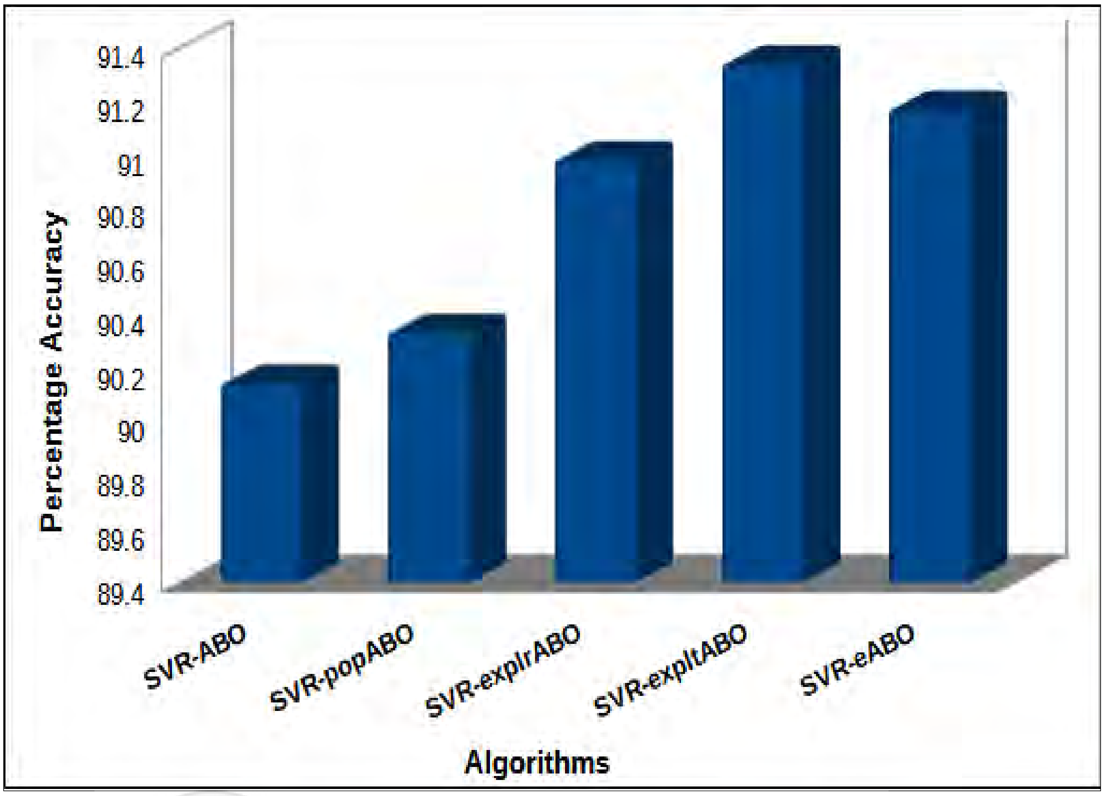

Figure 5.27: Comparison of Percentage Accuracy (PA) on Appliances dataset

5.3.1.6 CPU Execution Time

CPU execution represents the total execution time of the algorithms. Higher duration values, as seen in SVR-*eplt*ABO (331.3231) and SVR-*e*ABO (435.0065), indicate longer execution times. Lower duration values, exhibited by SVR-*pop*ABO (404.8427), SVR-ABO (358.9556), and SVR (0.3461), suggest faster execution times. It is observed that, though SVR-*e*ABO was able to record higher accuracy in terms of PA, yet it was achieved at the cost of higher execution time. Figure 5.28 depict a visual comparison of CPU time values of the developed algorithms.

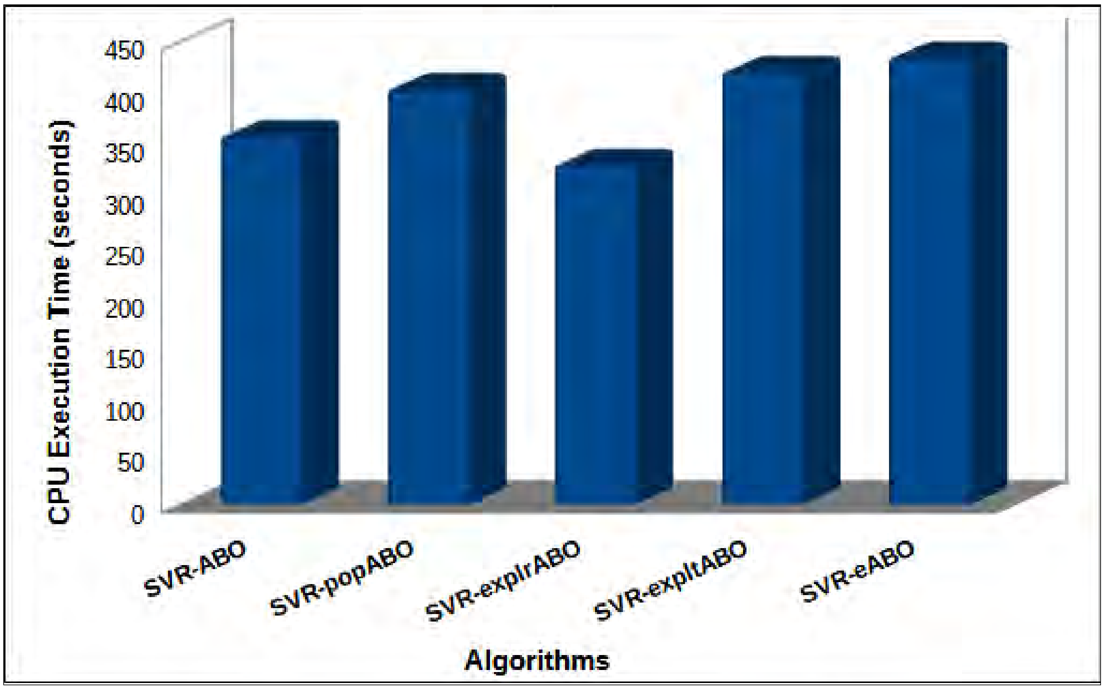

Figure 5.28: Comparison of CPU Execution Time on Appliances dataset

In summary, SVR-expltABO and SVR-eABO generally show better performance, with lower RMSE, MAPE, MAE values, higher R^2 and PA values indicates higher accuracy, lower forecasting errors, better goodness-of-fit, and a higher proportion of correctly predicted instances. SVR-ABO, SVR-*pop*ABO, and SVR have slightly lower performance metrics, suggesting slightly higher forecasting errors, lower goodness-of-fit, and a slightly lower percentage of correctly predicted instances. SVR-*explt*ABO and SVR-*e*ABO have longer execution times compared to SVR-*pop*ABO, SVR-ABO, and SVR.

5.3.2 Comparison of SVR-*e*ABO Against Benchmarks on Appliances Dataset

The performance of SVR-*e*ABO has been compared with various Support Vector Regression (SVR) variants, namely SVR-ABC, SVR-GA, SVR-PSO, SVR-CS, and was evaluated using several performance metrics. These metrics include Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), Percentage of Accurate Forecasting, and CPU execution time. Table 5.6 presents the comparative performance of the SVR-*e*ABO with benchmark algorithms.

Table 5.6

Comparison against eABO with Benchmarks on Appliances dataset

	SVR	**SVR-ABC**	**SVR-GA**	**SVR-PSO**	**SVR-CS**	**SVR-*e*ABO**
C	-	23.3082	0.4001	22.0227	48.3921	3.0768
Epsilon	-	0.0361	0.0836	0.0495	0.2108	0.0534
Gamma	-	0.0004	0.0302	0.0008	0.9053	0.0009
RMSE	455.9657	475.5483	444.0689	431.2986	480.6516	434.9863
MAPE	9.7568	9.0235	9.5019	9.1855	9.4894	8.8263
MAE	0.3742	0.3472	0.3205	0.3461	0.3701	0.3112
R^2	0.8127	0.8143	0.8166	0.8213	0.8154	0.8299
PA (%)	90.2432	90.9765	90.4981	90.8145	90.5106	91.1737
CPU Time	0.3461	363.0456	355.8537	412.4585	329.6102	435.0065

5.3.2.1 Root Mean Squared Error (RMSE)

Among the SVR variants, SVR-PSO achieved the lowest RMSE value of 431.2986, indicating superior performance in minimizing forecasting errors. It outperformed SVR-eABO, which had an RMSE of 434.9863, as well as SVR-GA with an RMSE of 444.0689, SVR-ABC with an RMSE of 475.5483, SVR-CS with an RMSE of 480.6516, and the classical SVR model with an RMSE of 455.9657. These results highlight that SVR-PSO is slightly better than the developed SVR-*e*ABO algorithm in terms of minimizing forecasting errors based on RMSE compared to the other SVR variants and the classical SVR model on Appliances dataset. Figure 5.29 shows a visual comparative performance on RMSE metric of all the benchmarks and the developed SVR-*e*ABO algorithm.

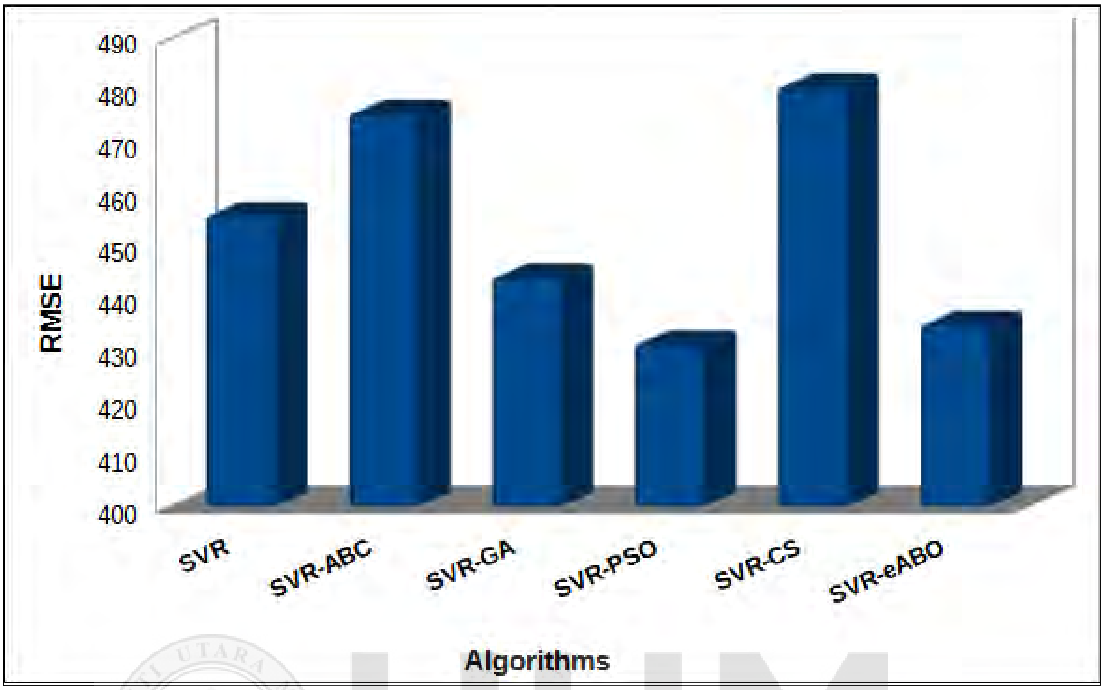

Figure 5.29: Comparison of RMSE based on Benchmarks on Appliances dataset

5.3.2.2 Mean Absolute Percentage Error (MAPE)

In terms of MAPE, SVR-*e*ABO achieved the lowest value of 8.8263, followed by SVR-ABC with a MAPE of 9.0235, SVR-GA with a MAPE of 9.5019, SVR-PSO with a MAPE of 9.1855, SVR-CS with a MAPE of 9.4894, and the classical SVR model with a MAPE of 9.7568. These results indicate that SVR-*e*ABO exhibited the smallest average relative deviation from the true values in percentage terms, indicating its superior performance in accuracy. Figure 5.30 shows a visual comparative performance on MAPE metric of all the benchmarks and the developed SVR-*e*ABO algorithm.

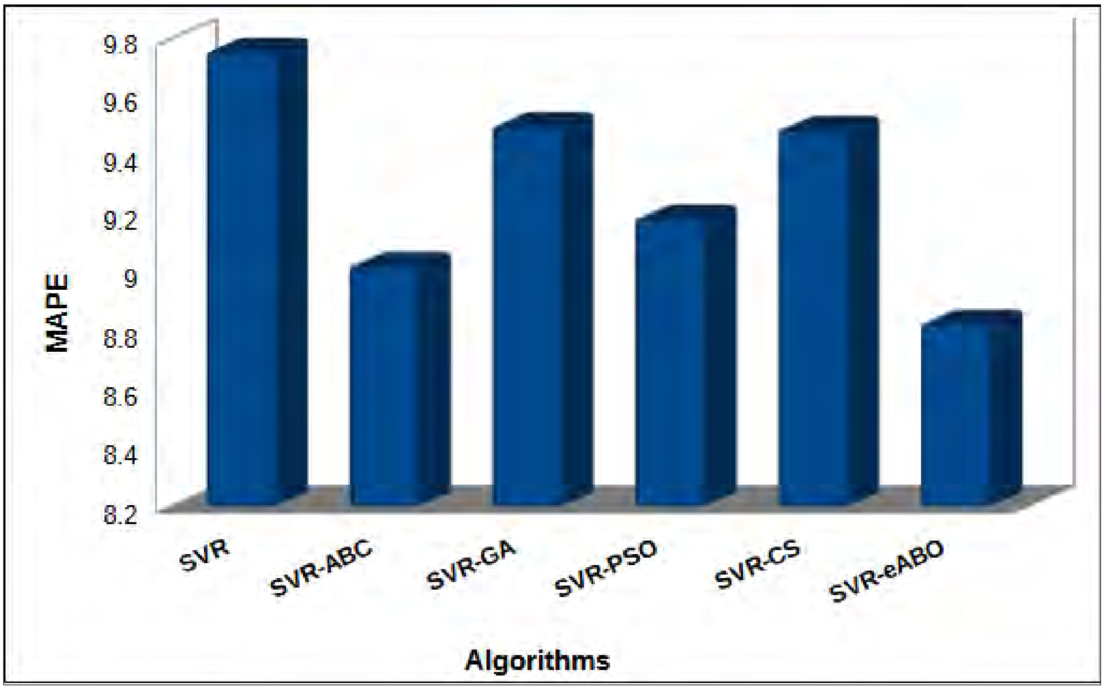

Figure 5.30: Comparison of RMSE based on Benchmarks on Appliances dataset

5.3.2.3 Mean Absolute Error (MAE)

SVR-*e*ABO achieved the lowest MAE value of 0.3112, followed by SVR-GA with a MAE of 0.3205, SVR-PSO with a MAE of 0.3461, SVR-ABC with a MAE of 0.3472, SVR-CS with a MAE of 0.3701, and the classical SVR model with a MAE of 0.3742. These results suggest that SVR-*e*ABO minimized the absolute forecasting errors more effectively compared to the other SVR variants and the classical SVR model. Figure 5.31 shows a visual comparative performance on MAE metric of all the benchmarks and the developed SVR-*e*ABO algorithm.

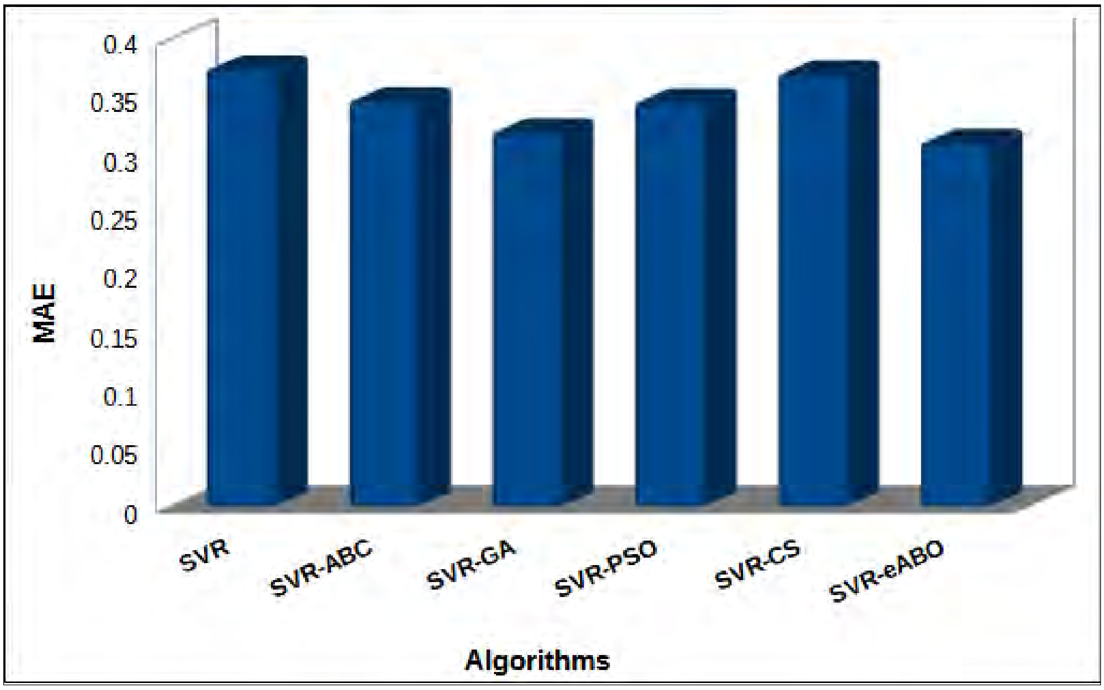

Figure 5.31: Comparison of MAE based on Benchmarks on Appliances dataset

5.3.2.4 Coefficient of Determination (R^2)

SVR-*e*ABO achieved an R^2 value of 0.8299, followed by SVR-PSO with an R^2 of 0.8213, SVR-GA with an R^2 of 0.8166, SVR-CS with an R^2 of 0.8154, SVR-ABC with an R^2 of 0.8143, and the classical SVR model with an R^2 of 0.1827. These results indicate that SVR-*e*ABO exhibited the highest ability to explain the variance in the target variable compared to the other SVR variants and the classical SVR model. Figure 5.32 shows a visual comparative performance on R^2 metric of all the benchmarks and the developed SVR-eABO algorithm.

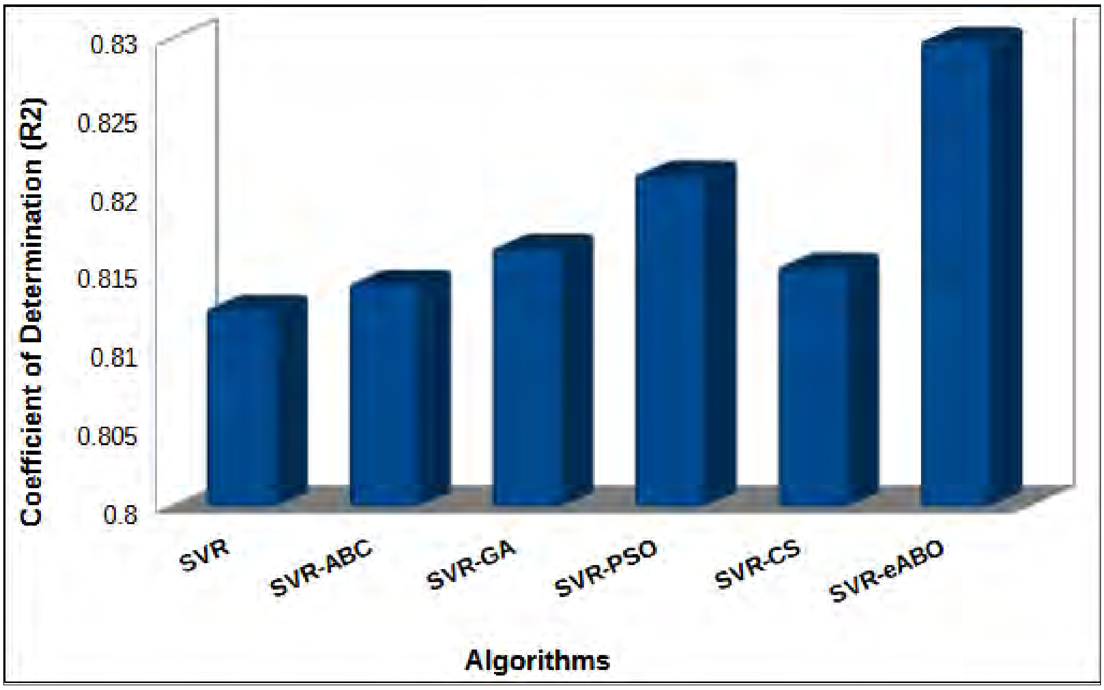

Figure 5.32: Comparison of R^2 based on Benchmarks on Appliances dataset

5.3.2.5 Percentage of Accurate (PA)

SVR-eABO achieved the highest accuracy with a PA of 91.1737, followed by SVR-ABC with a PA (%) of 90.9765, SVR-GA with a PA of 90.4981, SVR-CS with a PA of 90.5106, SVR-PSO with a PA of 90.8145, and the classical SVR model with a PA of 90.2432. These results indicate that SVR-*e*ABO produced the highest proportion of accurate forecasting among the SVR variants and the classical SVR model. Figure 5.33 shows a visual comparative performance on R^2 metric of all the benchmarks and the developed SVR-*e*ABO algorithm.

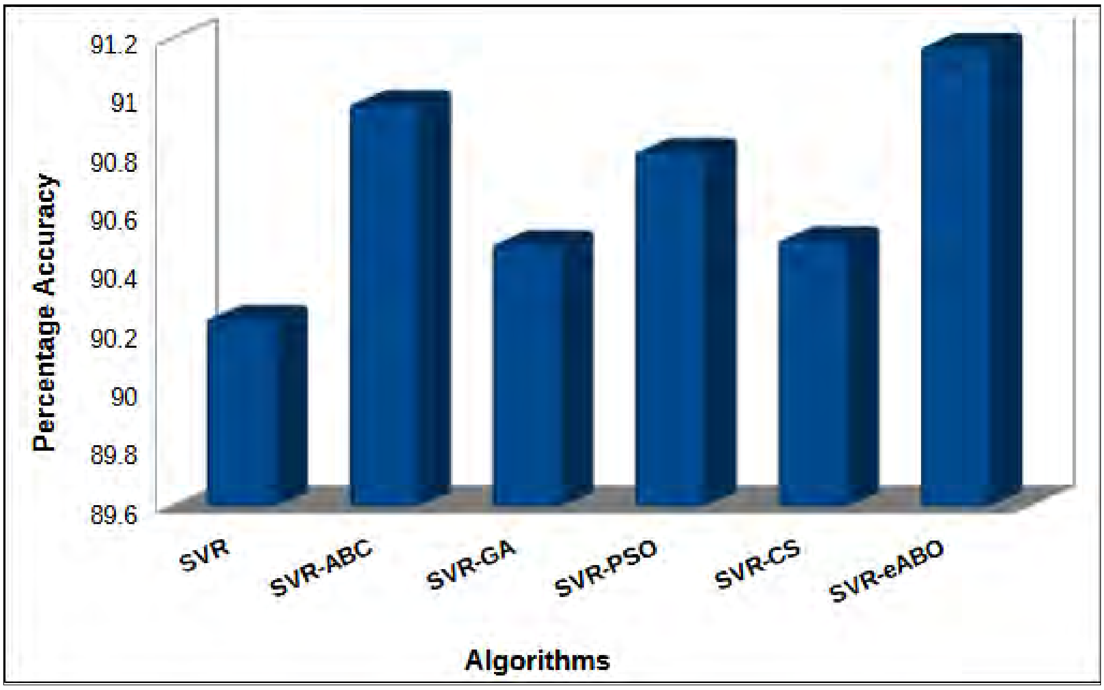

Figure 5.33: Comparison of PA based on Benchmarks on Appliances dataset

5.3.2.6 CPU Execution Time

Among the SVR variants, classical SVR reported the shortest CPU execution time of 0.3461 followed SVR-CS with CPU execution time of 329.6102, followed by SVR-ABC with a CPU execution time of 402.5136, SVR-GA with a duration of 421.2799, SVR-PSO with a duration of 448.6087, and SVR-*e*ABO with a duration of 518.4234. These results indicate that classical SVR exhibited the highest computational efficiency among the SVR variants considered. Figure 5.34 shows a visual comparative performance on CPU execution time as evaluation metric of all the benchmarks and the developed SVR-eABO algorithm.

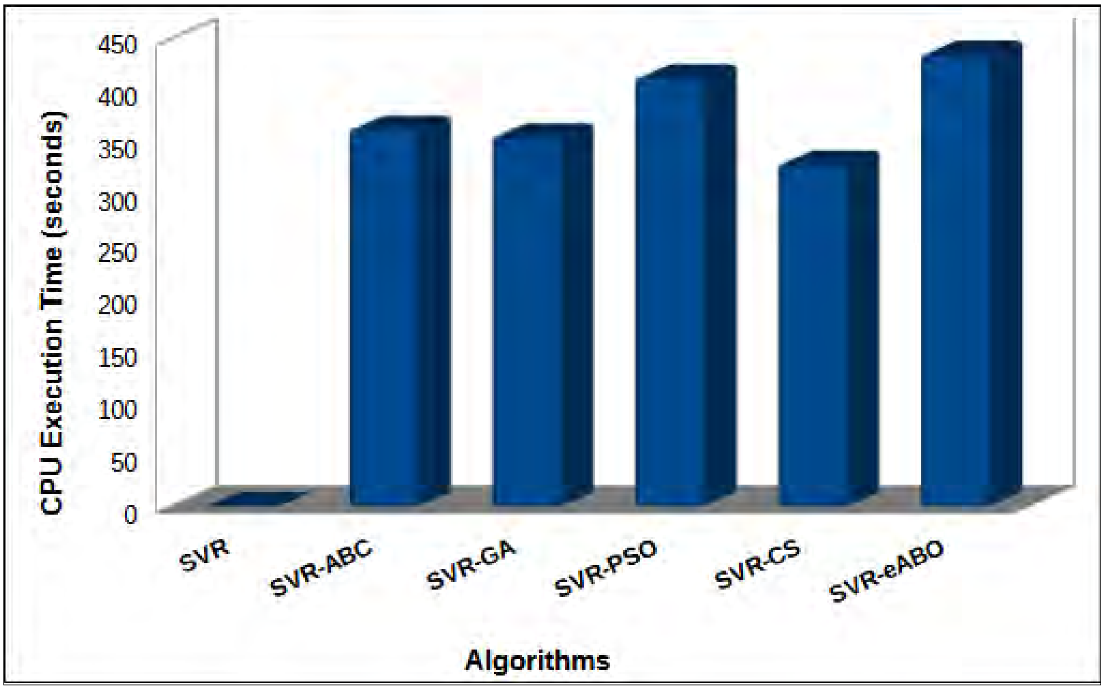

Figure 5.34: Comparison of CPU Time (against Benchmarks) on Appliances dataset

In summary, the comparative analysis of the SVR variants reveals that SVR-*e*ABO consistently outperformed the other models in terms of RMSE, MAPE, MAE, R^2, and PA (%). These results indicate that SVR-eABO exhibited superior performance in minimizing forecasting errors, and higher forecasting accuracy compared to the other SVR variants and the classical SVR model. Additionally, classical SVR demonstrated the highest computational efficiency among the SVR variants in terms of CPU execution time.

5.4 Panama dataset

This section presents a comparative analysis evaluating the performance of several enhanced algorithms on the Panama dataset. Furthermore, the performance of the final algorithm, SVR-*e*ABO, was assessed in comparison to the hybrid algorithms developed at various stages, utilizing the same dataset. The subsequent results delineate the performance improvements attained at each stage and provide a comparative evaluation against the culminating SVR-*e*ABO approach. Table 5.7 provides performance values of the comparisons.

5.4.1 Comparison Between Developed Algorithms on Panama dataset

Table 5.7

Comparison against developed algorithms on Panama dataset

	SVR-ABO	**SVR-*pop*ABO**	**SVR-*explr*ABO**	**SVR-*explt*ABO**	**SVR-*e*ABO**
C	0.5687	888.1198	888.1198	478.9911	560.8820
Epsilon	0.4130	0.0061	0.0061	0.0072	0.0073
Gamma	0.0862	0.0381	0.0381	0.0436	0.0394
RMSE	**1406.6100**	1425.9649	1429.3005	1423.1092	1419.3501
MAPE	3.5081	3.4897	3.4882	3.4792	**3.4765**
MAE	1021.1916	1017.0317	1016.1864	1039.0039	1012.9322
R^2	0.8509	0.8495	0.8493	0.8500	0.8500
PA (%)	96.4919	96.5103	96.5118	96.5208	**96.5235**
CPU Time	**621.2687**	712.4568	798.6220	681.2687	786.2687

5.4.1.1 Root Mean Square Error (RMSE)

RMSE measures the average magnitude of forecasting errors. A lower RMSE value indicates better accuracy and smaller forecasting errors, suggesting that the predicted values are closer to the actual values. Among the algorithms, SVR-ABO has the lowest RMSE value (1406.6100), followed closely by SVR-*e*ABO (1419.3501). This suggests that both algorithms have better accuracy and smaller forecasting errors compared to the other algorithms. On the other hand, SVR-*pop*ABO (1425.9649), SVR-*explr*ABO (1429.3005), SVR-*explt*ABO (1423.1092), and SVR (1425.1367) have higher RMSE values, indicating larger forecasting errors. These algorithms demonstrate more difficulty to accurately predict the target variable. Figure 5.35 depict a visual comparison of RMSE values of the developed algorithms.

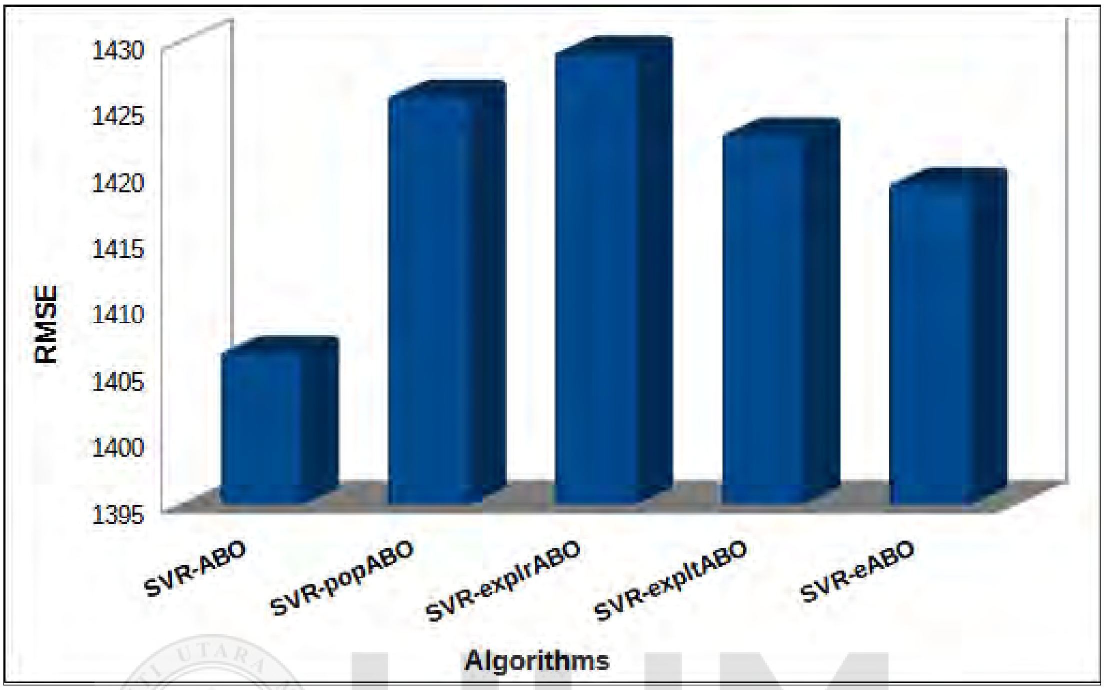

Figure 5.35: Comparison of RMSE on Panama dataset

5.4.1.2 Mean Absolute Percentage Error (MAPE)

MAPE measures the average percentage deviation between the predicted and actual values. A lower MAPE value indicates better accuracy and smaller forecasting errors in percentage terms. SVR-ABO (3.5081), SVR-*e*ABO (3.4765), and SVR-*pop*ABO (3.4897) have relatively lower MAPE values, suggesting better accuracy and smaller forecasting errors in percentage terms. SVR-*explr*ABO (3.4882), SVR-*explt*ABO (3.4792), and SVR (3.5260) have slightly higher MAPE values, indicating larger forecasting errors in percentage terms. Figure 5.36 depict a visual comparison of MAPE values of the developed algorithms.

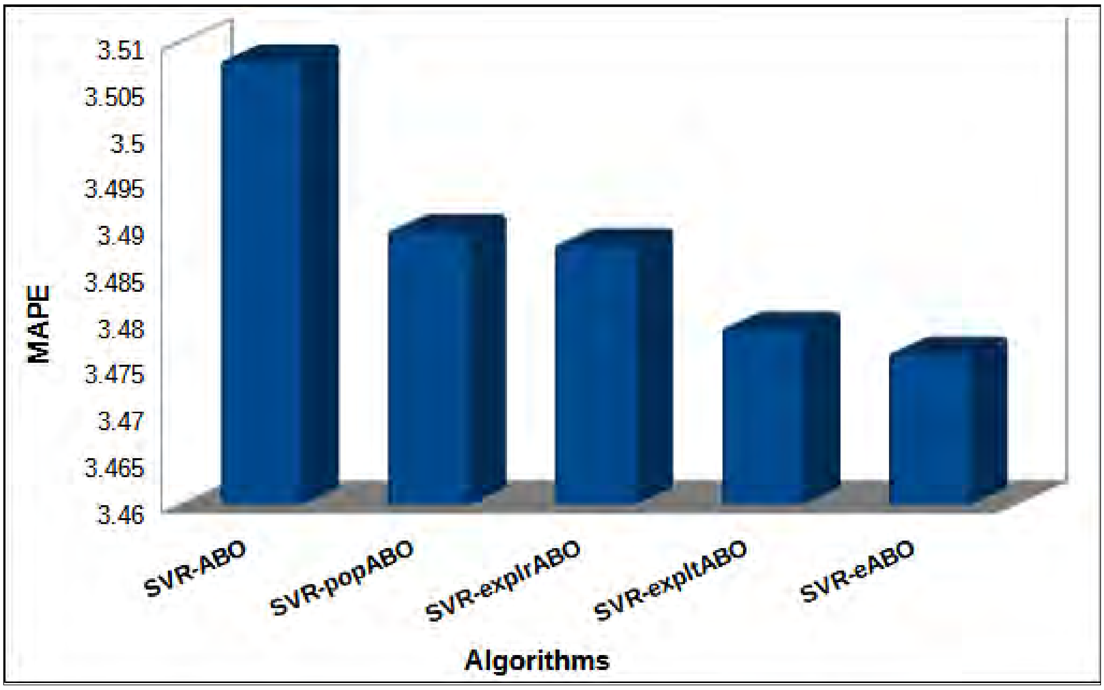

Figure 5.36: Comparison of MAPE on Panama dataset

5.4.1.3 Mean Absolute Error (MAE)

MAE measures the average magnitude of forecasting errors without considering their direction. A lower MAE value indicates better accuracy and smaller forecasting errors. SVR-*e*ABO (1012.9322) has the lowest MAE value, followed by SVR-ABO (1021.1916) and SVR (1025.3182). These algorithms exhibit better accuracy and smaller forecasting errors compared to the other algorithms. SVR-*pop*ABO (1017.0317), SVR-*explr*ABO (1016.1864), and SVR-*explt*ABO (1039.0039) have slightly higher MAE values, indicating larger forecasting errors. Figure 5.37 depict a visual comparison of MAE values of the developed algorithms.

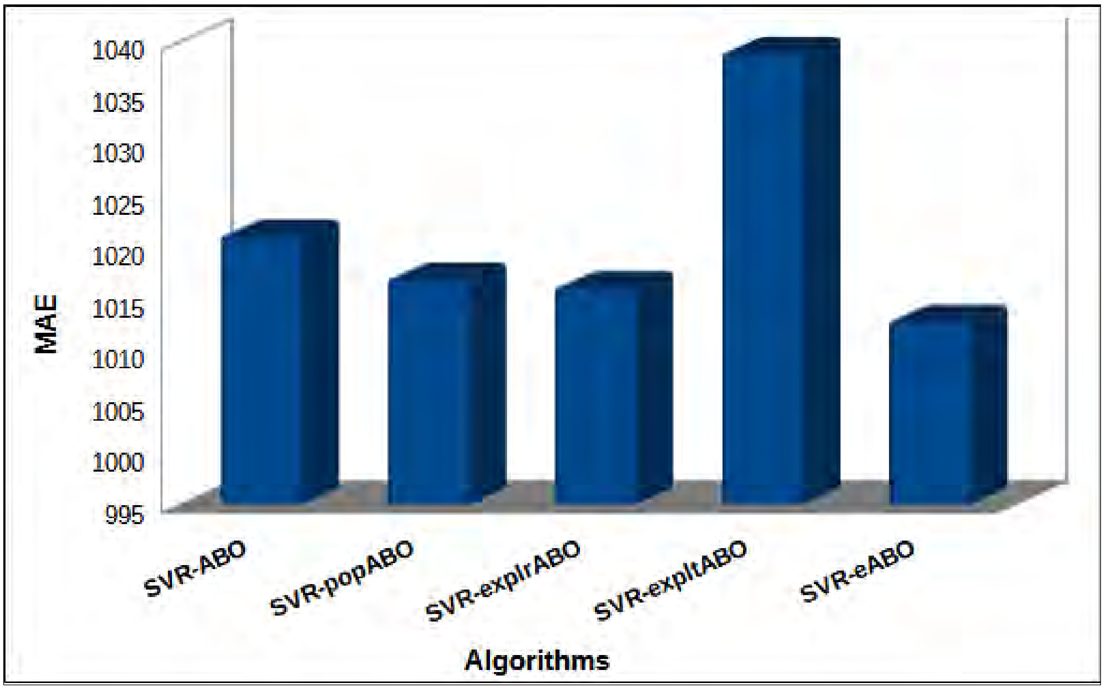

Figure 5.37: Comparison of MAE on Panama dataset

5.4.1.4 Coefficient of Determination (R^2):

R^2 measures the proportion of the variance in the dependent variable that is explained by the independent variables. A higher R^2 value indicates a better fit to the data. SVR-ABO (0.508467) has the highest R^2 value among the algorithms, suggesting a better fit to the data and a higher proportion of variance explained. SVR-*e*ABO (0.4995) and SVR (0.49543381) also exhibit moderate R^2 values, indicating reasonable fits to the data. SVR-*pop*ABO (0.4949), SVR-*explr*ABO (0.4925), and SVR-*explt*ABO (0.4968684) have slightly lower R^2 values compared to the other algorithms. Figure 5.38 depict a visual comparison of R^2 values of the developed algorithms.

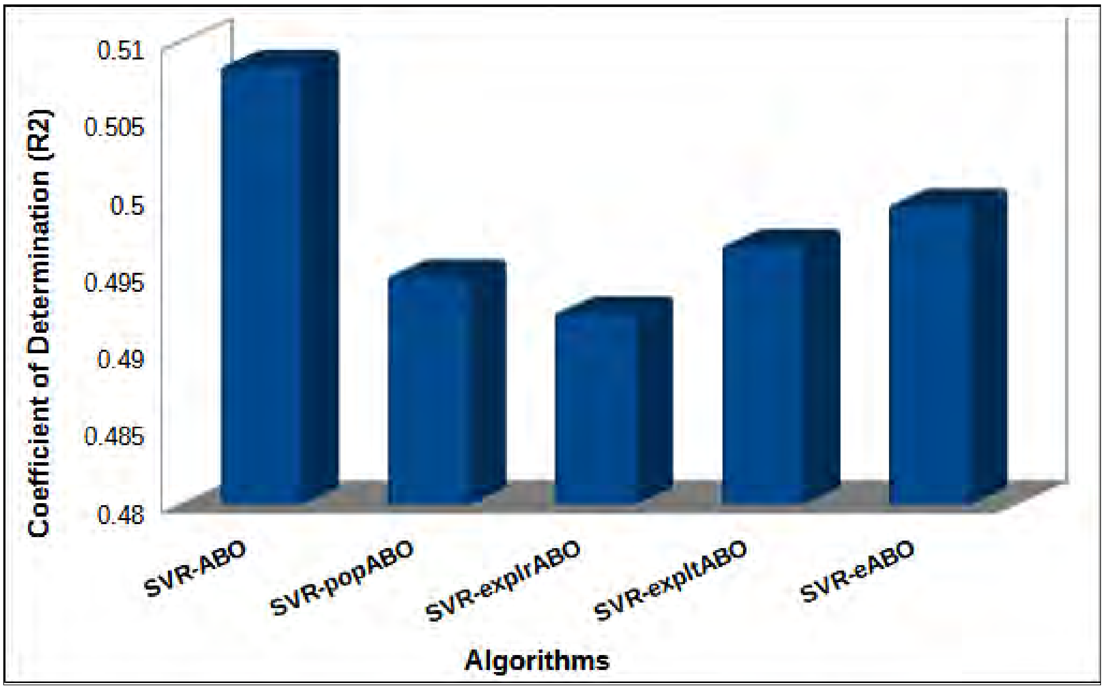

Figure 5.38: Comparison of R^2 on Panama dataset

5.4.1.5 Percentage of Accuracy (PA):

The analysis of the forecasting accuracy values revealed that all the evaluated algorithms demonstrated similar performance, with values ranging from 96.4740% to 96.5235%. These results indicate that the algorithms are capable of making accurate forecasting with a high degree of consistency. The narrow range of forecasting accuracy values suggests that the algorithms have comparable capabilities in capturing the underlying patterns in the dataset and generating accurate forecasting. However, upon closer examination, SVR-*e*ABO consistently outperformed the other algorithms by achieving the highest forecasting accuracy of 96.5235%. This indicates that SVR-*e*ABO exhibits superior predictive capabilities compared to both the classical SVR model and the other SVR variants considered in this study. The incorporation of swarm-based ABO in SVR-*e*ABO contributes to its improved predictive accuracy by enhancing the optimization process. The ABO algorithm enables SVR-eABO to effectively explore the solution space and find optimal parameter settings, leading to enhanced forecasting accuracy. The superior performance of SVR-*e*ABO in terms of forecasting accuracy suggests its potential as a preferred choice for predictive modeling tasks. Figure 5.39 depict a visual comparison of PA values of the developed algorithms.

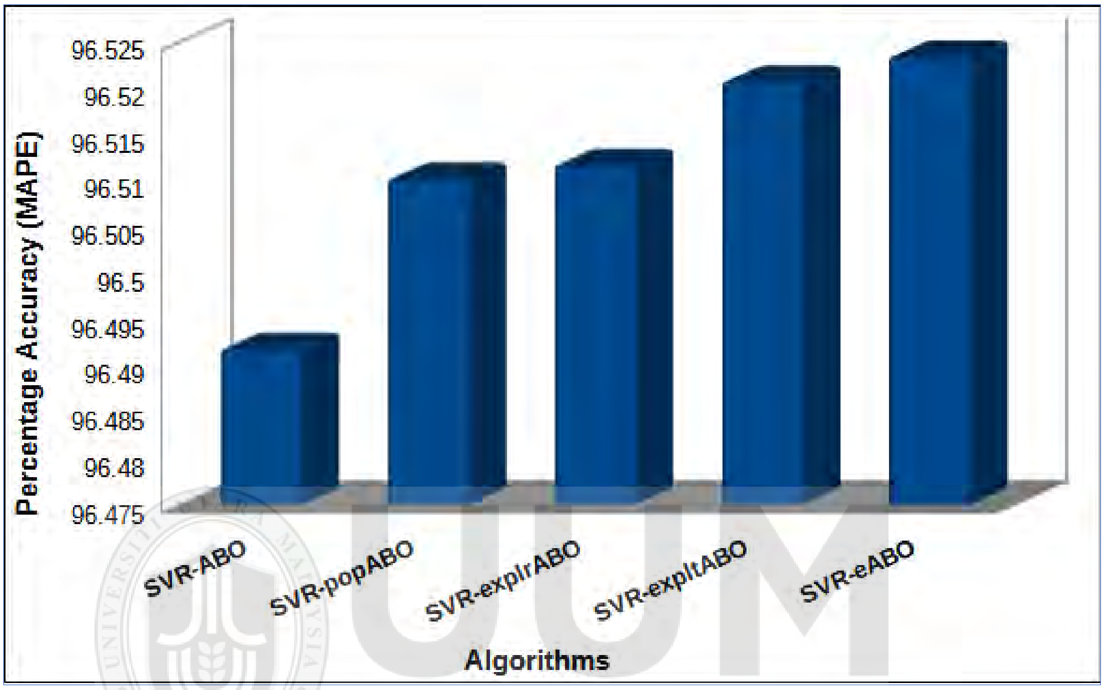

Figure 5.39: Comparison of Percentage Accuracy (PA) on Panama dataset

5.4.1.6 CPU Execution Time

SVR demonstrates the shortest execution time among all the algorithms, with a duration of 0.06721. This indicates that SVR is highly efficient in terms of CPU execution. However, despite its longer duration of 621.2687, SVR-ABO was able to achieve a higher percentage accuracy (PA) of 96.4919 compared to basic SVR. SVR-*pop*ABO exhibits a further increase in duration compared to both SVR and SVR-ABO, with a value of 712.4568. Although it incurs a higher computational overhead, SVR-*pop*ABO still manages to achieve a slightly higher PA of 96.5103 compared to SVR-ABO. On the other hand, SVR-*explr*ABO has a significantly longer duration of 798.6220 compared to the previous algorithms. Despite this, SVR-*explr*ABO was able to record a PA value of 96.5118. In contrast, SVR-*explt*ABO demonstrates a slightly shorter duration compared to SVR-*explr*ABO, with a value of 681.2687. Interestingly, SVR-*explt*ABO achieves a slightly higher PA of 96.5208 compared to SVR-*explr*ABO. Finally, SVR-*e*ABO exhibits a significantly longer duration of 786.2687 compared to SVR. However, the algorithm still manages to achieve a slightly higher PA value of 96.5235. Figure 5.40 depict a visual comparison of CPU execution time in seconds of the developed algorithms.

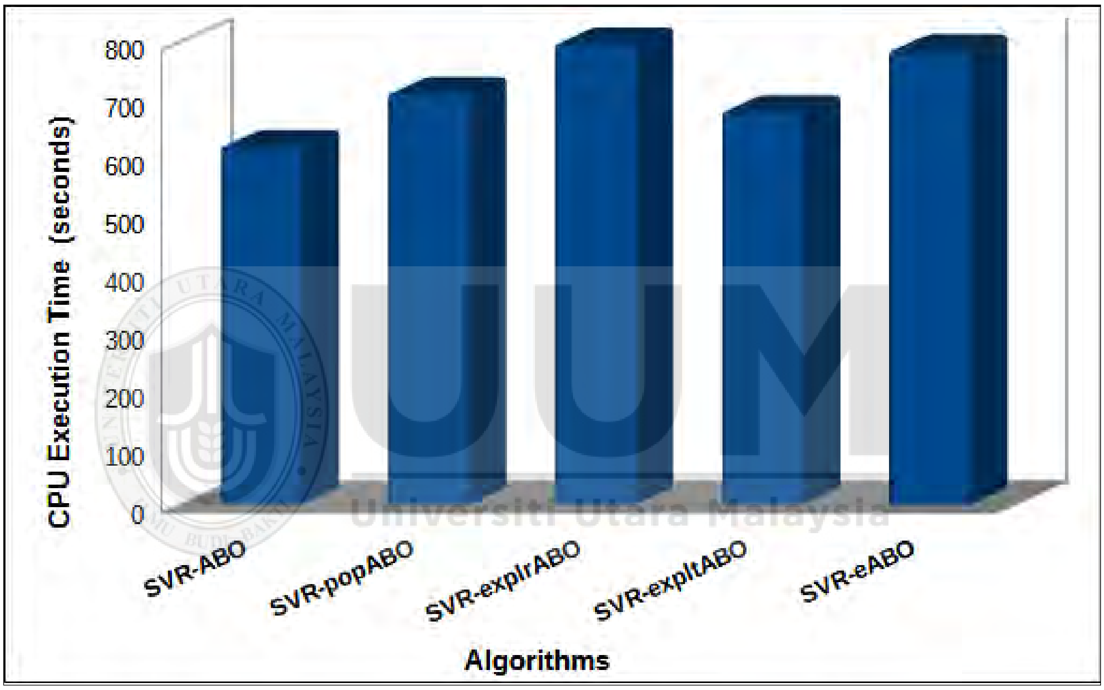

Figure 5.40: Comparison of CPU Execution Time on Panama dataset

Conclusively, SVR-*e*ABO generally demonstrates better accuracy and smaller forecasting errors compared to the other algorithms, as indicated by lower values in RMSE, MAPE, and MAE. SVR-ABO has the highest R^2 value, suggesting a better fit to the data and a higher proportion of variance explained. The other algorithms (SVR, SVR-*pop*ABO, SVR-*explr*ABO, SVR-*explt*ABO) generally exhibit slightly higher RMSE, MAPE, and MAE values, indicating larger forecasting errors and reduced accuracy compared to SVR-*e*ABO.

5.4.2 Comparison of SVR-*e*ABO Against Benchmarks on Panama Dataset

The performance of SVR-*e*ABO has been compared with various Support Vector Regression (SVR) variants, namely SVR-ABC, SVR-GA, SVR-PSO, SVR-CS, and was evaluated using several performance metrics. These metrics include Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), Percentage of Accurate Forecasting, and CPU execution time. Table 5.8 presents the comparative performance of the SVR-*e*ABO with benchmark algorithms.

Table 5.8

Comparison against eABO with Benchmarks on Panama dataset

	SVR	SVR-ABC	SVR-GA	SVR-PSO	SVR-CS	SVR-*e*ABO
C	-	6.5927	0.0472	0.1124	17.0917	560.8820
Epsilon	-	0.16215	0.0710	0.3149	0.0482	0.0073
Gamma	-	0.0192	0.0636	0.4443	0.0390	0.0394
RMSE	1425.1367	1444.8945	1418.0946	1454.0075	1430.9824	1419.3501
MAPE	3.5260	3.5544	3.6329	3.6094	3.5902	3.4765
MAE	1025.3182	1036.1330	1059.3854	1052.4854	1031.5462	1012.9322
R^2	0.4954	0.5004	0.4748	0.4814	0.4853	0.4995
PA (%)	96.4740	96.4456	96.3671	96.3906	96.4098	**96.5235**
CPU Time	0.06721	611.4653	624.9735	774.0880	683.8492	786.2687

5.4.2.1 Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) analysis revealed that SVR-GA outperformed SVR-ABC with a slightly lower RMSE value of 1418.0946 compared to 1444.8945. Despite a higher RMSE of 1454.0075, SVR-PSO competes closely with SVR-GA in predictive accuracy. SVR-CS maintained a competitive edge with an RMSE of

1430.9824, while SVR-*e*ABO, as the algorithm that has been benchmarked with these just mentioned algorithms, showcased robust predictive capabilities with an RMSE of 1419.3501 as demonstrated in figure 5.41.

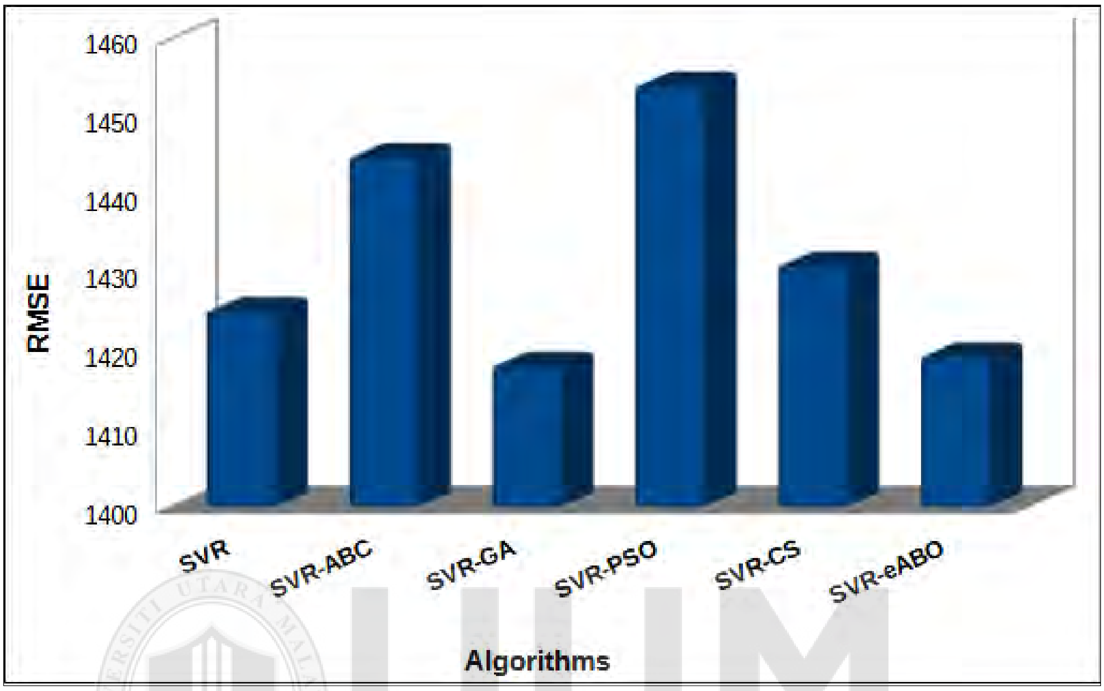

Figure 5.41: Comparison of RMSE based on Benchmarks on Panama dataset

5.4.2.2 Mean Absolute Percentage Error (MAPE)

SVR-eABO, the algorithm developed in this study, showcased exceptional accuracy in comparison to the benchmarked algorithms. It achieved a notably low Mean Absolute Percentage Error (MAPE) value of 3.4765, outperforming SVR-ABC (MAPE = 3.5260), SVR-GA (MAPE = 3.6329), SVR-PSO (MAPE = 3.6094), and SVR-CS (MAPE = 3.5902). This indicates that SVR-*e*ABO yielded highly precise forecasting with minimal deviation from the actual values.

Furthermore, SVR-*e*ABO demonstrated a superior performance across the board, displaying the lowest MAPE value of 3.4765. Following SVR-*e*ABO, SVR-CS exhibited the next best performance with a MAPE value of 3.5902, implying a relatively higher but still commendable level of accuracy. SVR-GA, although slightly less accurate, achieved a MAPE value of 3.6329.

The results highlight the effectiveness of SVR-*e*ABO in minimizing forecasting errors and enhancing the overall accuracy of the algorithm. This suggests the potential of SVR-*e*ABO for accurate forecasting and its superiority over the other benchmarked algorithms in the specific context of this study.

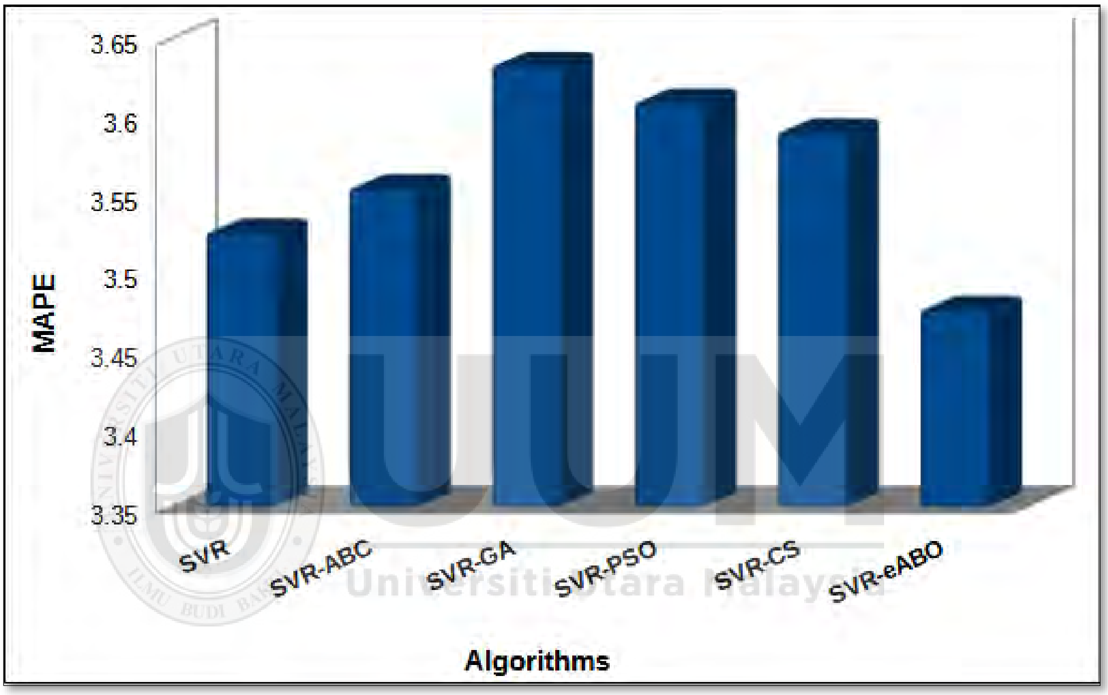

Figure 5.42: Comparison of MAPE based on Benchmarks on Panama dataset

5.4.2.3 Mean Absolute Error (MAE)

SVR-eABO demonstrated exceptional performance with a MAE value of 1012.9322. This indicates that SVR-*e*ABO exhibited a lower error magnitude and higher accuracy in comparison to the benchmarked algorithms, namely SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS. Specifically, SVR-ABC recorded a MAE value of 1036.1330, implying a relatively higher error magnitude compared to SVR-eABO. Similarly, SVR-GA and SVR-PSO displayed slightly higher error magnitudes, suggesting a reduced level of accuracy in their forecasting. These findings reinforce the superior accuracy and improved forecasting capability of SVR-*e*ABO over the other benchmarked algorithms, as evident from its lower MAE value. SVR-*e*ABO emerges as a promising algorithm for minimizing error and enhancing accuracy in the specific context of this study. Figure 5.43 shows the graphical representation of the algorithms' performance.

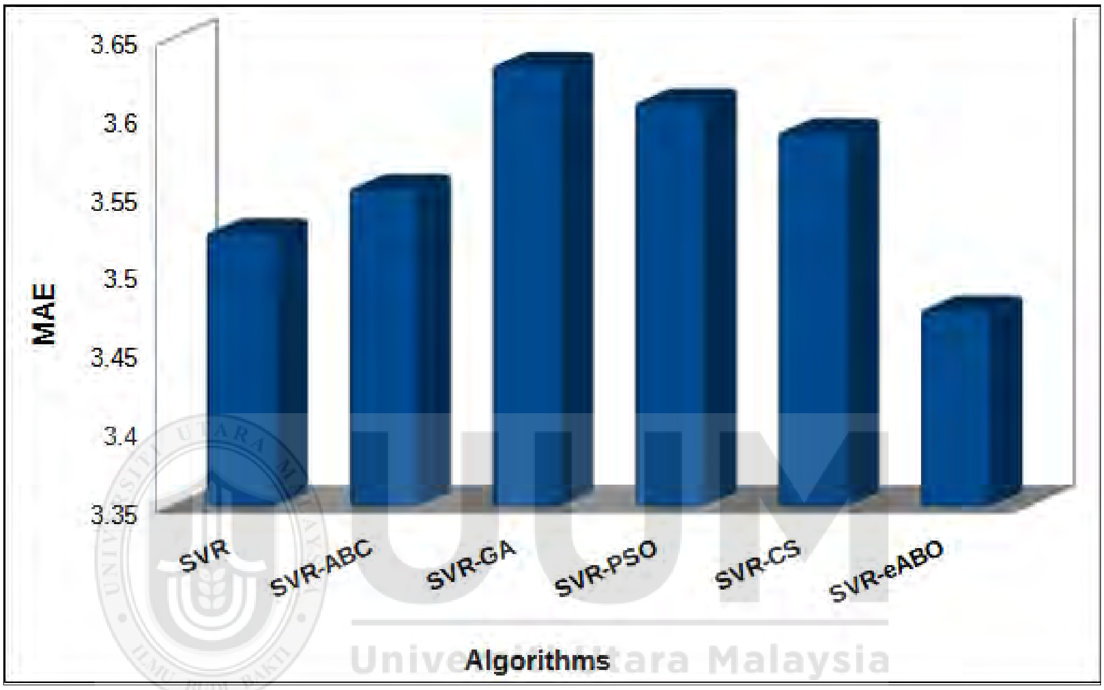

Figure 5.43: Comparison of MAE based on Benchmarks on Panama dataset

5.4.2.4 Coefficient of Determination (R^2)

The analysis of the Coefficient of Determination (R^2) revealed distinct levels of predictive power among the algorithms, namely SVR-ABC, SVR-GA, SVR-PSO, SVR-CS, and SVR-*e*ABO. Notably, SVR-*e*ABO emerged as the frontrunner with an R^2 value of 0.4995, indicating strong predictive abilities and high explanatory power.

Comparatively, the conventional SVR algorithm (SVR) achieved an R^2 value of 0.4954, suggesting a relatively lower level of predictive power. SVR-ABC exhibited a slightly higher R^2 value of 0.5004, indicating better predictive performance than the conventional SVR algorithm. SVR-GA and SVR-PSO demonstrated R^2 values of

0.4748 and 0.4814, respectively, indicating a moderate level of predictive power. SVR-CS achieved an R^2 value of 0.4853, indicating a similar level of predictive performance as graphically presented in figure 5.44. Overall, these findings underscore the superior predictive capabilities and higher explanatory power of SVR-*e*ABO, as reflected by its higher R^2 value when compared to the other algorithms considered in this analysis.

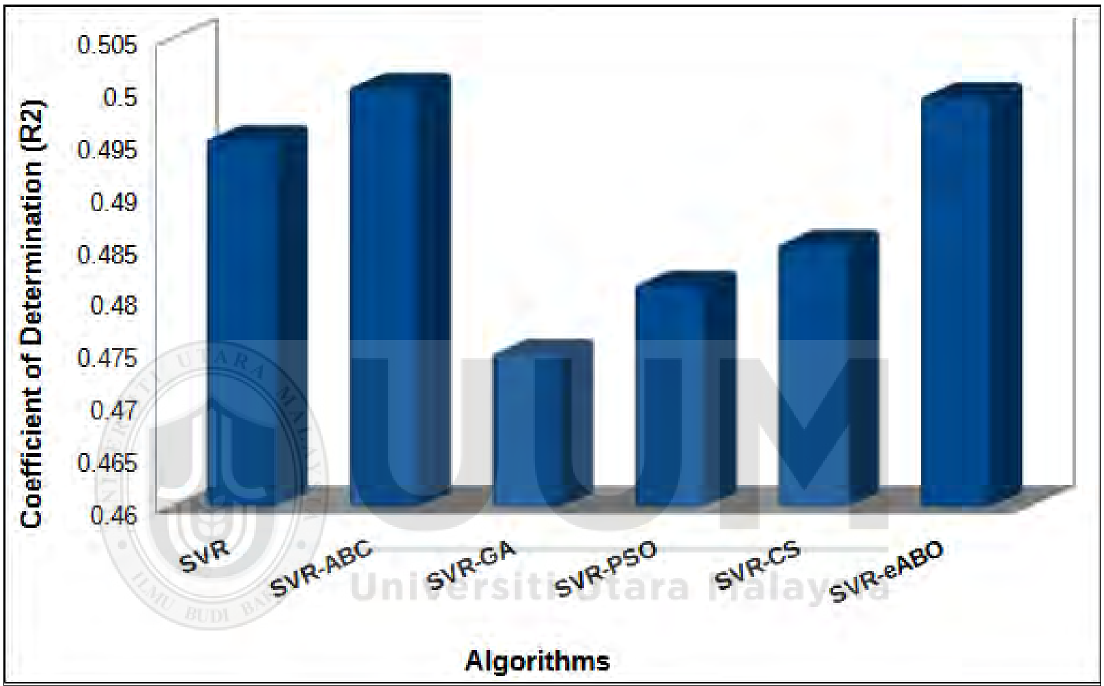

Figure 5.44: Comparison of R^2 based on Benchmarks on Panama dataset

5.4.2.5 Percentage Accuracy (PA)

Upon assessing the Percentage Accuracy (PA), SVR-*e*ABO exhibited superior accuracy in making forecasting, achieving a value of 96.5235. This surpassed the forecasting accuracy of SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS. Specifically, the conventional SVR algorithm (SVR) achieved a forecasting accuracy of 96.4740, SVR-ABC attained 96.4456, SVR-GA achieved 96.3671, SVR-PSO reached 96.3906, and SVR-CS obtained 96.4098.

The higher PA % of SVR-*e*ABO (96.5235) suggests greater precision and accuracy in predicting outcomes compared to the other algorithms. These results, as depicted in figure 5.45, highlight the superior predictive performance of SVR-*e*ABO in terms of forecasting accuracy, reinforcing its efficacy as a reliable model for generating accurate forecasting.

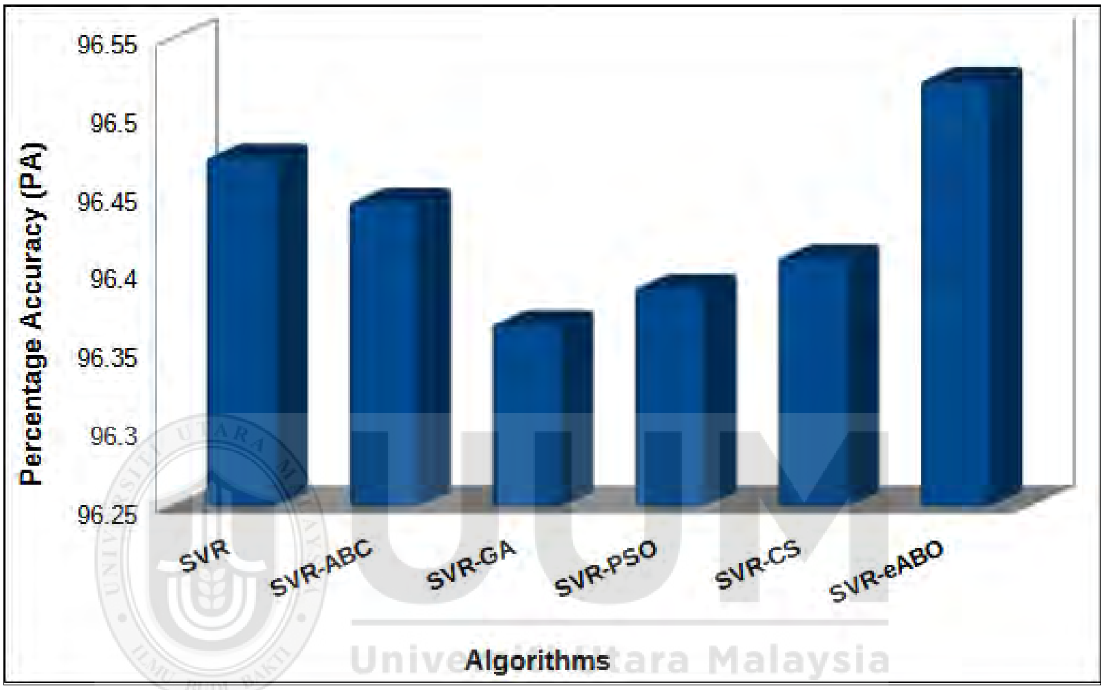

Figure 5.45: Comparison of Percentage Accuracy based on Benchmarks on Panama dataset

5.4.2.6 CPU Execution Time

The SVR algorithm has a CPU execution time of 0.06721. This indicates that it is the fastest algorithm among the considered variants, with the lowest execution time. However, it is important to note that its forecasting accuracy, as mentioned earlier, is also the lowest. The SVR-ABC variant has a CPU execution time of 611.4653. This suggests that it takes significantly longer to execute compared to SVR. Despite the increased execution time, SVR-ABC achieves a higher forecasting accuracy, indicating a trade-off between computational cost and accuracy.

On other hand, SVR-GA variant has a CPU execution time of 624.9735. This is similar to the CPU execution time of SVR-ABC, indicating that both variants require a comparable number of computational resources. SVR-GA also achieves a higher forecasting accuracy, suggesting that the additional execution time may be justified by improved accuracy. The SVR-PSO variant has a CPU execution time of 774.0880. This indicates a further increase in execution time compared to SVR-ABC and SVR-GA. The SVR-CS variant has a CPU execution time of 683.8492. This execution time is similar to that of SVR-ABC and SVR-GA, indicating comparable computational requirements. SVR-CS achieves a slightly lower forecasting accuracy compared to SVR-PSO but still outperforms SVR and SVR-ABC in terms of accuracy. The SVR-*e*ABO algorithm has a CPU execution time of 786.2687. This is the highest execution time among the considered variants as can be seen from figure 5.46, indicating that it requires the most computational resources. However, SVR-*e*ABO also achieves the highest forecasting accuracy, suggesting that the additional computational cost may be justified by its superior performance.

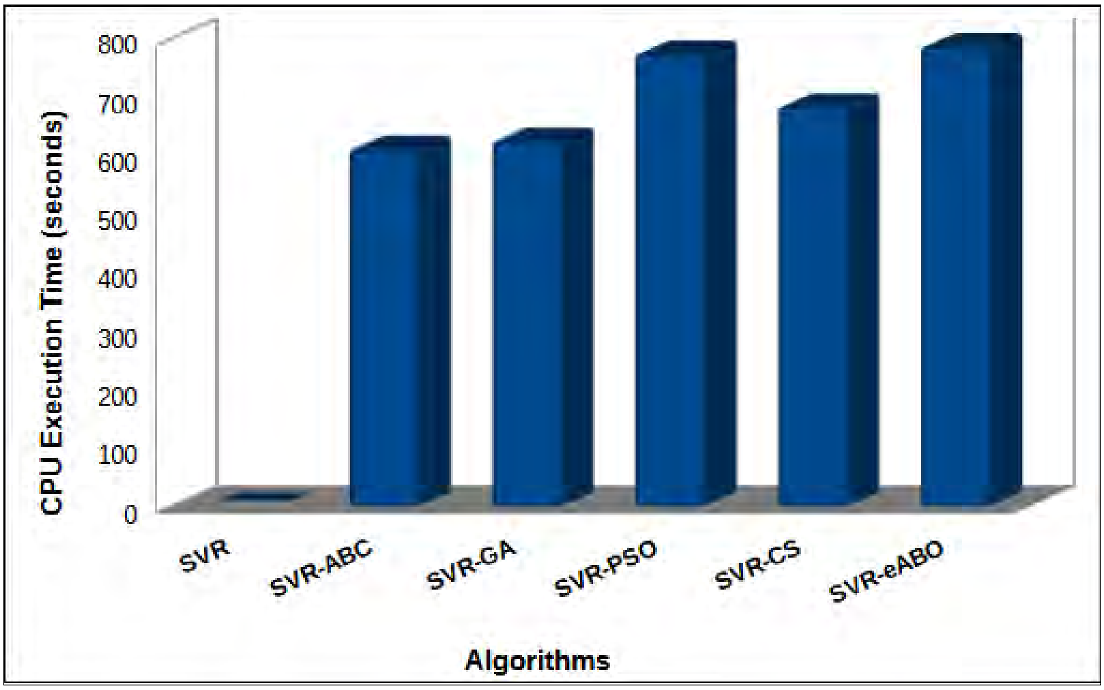

Figure 5.46: Comparison of CPU Time (against Benchmarks) on Panama dataset

In summary, the benchmark algorithms (SVR-ABC, SVR-GA, SVR-PSO, and SVR-CS) generally exhibit higher CPU execution times compared to the base SVR algorithm. However, they also achieve higher forecasting accuracies, indicating improved performance. Among the benchmark algorithms, SVR-*e*ABO has the highest execution time but also the highest forecasting accuracy. This suggests that SVR-*e*ABO may provide a good trade-off between computational cost and predictive performance, making it a promising algorithm for applications where accuracy is crucial and computational resources are available.

5.5 Comparison Between Algorithms on Standard Optimisation Functions

In this section, the developed algorithm has been tested on sixteen (16) selected optimisation functions as described in section 3.5.2. Some of these optimisation functions have single local optima (unimodal), while others have several optima (multi-modal) with unique global optima. The developed algorithm was run for hundred (100) number of independent runs of which the mean and standard deviation were recorded. This is to ensure that performance of each algorithm is adequately represented. The test was performed with a hundred (100) buffaloes in three (3) dimension space representing the number of dimensions of problem at hand. While each algorithm was run within the range of thirty (30) function evaluations as suggested in (Long et al., 2018; Mirjalili et al., 2014). The performance of the enhanced ABO (SVR-*e*ABO) algorithm based on cumulative enhancements performed at all three stages (Population Initialisation, Exploration, and Exploitation) has been compared with the SVR-PSO, SVR-ABC, SVR-CS and SVR-GA. The result obtained is as presented in table 5.9 and table 5.10 respectively.

5.5.1 Performance of Developed algorithms on Standard Optimisation Functions

Table 5.9

Comparison of developed algorithms on SOF

			SVR-*pop*ABO		SVR-*explr*ABO		SVR-*explt*ABO		SVR- eABO	
Function	**C**	**GO**	**Mean**	**Stdv**	**Mean**	**Stdv**	**Mean**	**Stdv**	**Mean**	**Mean**
F_1	M	0	5.47E-05	3.09E-05	**8.10E-08**	1.17E-07	1.65E-03	8.04E-03	8.31E-08	8.31E-08
F_2	U	0	4.087E-05	3.92E-05	**2.09E-09**	3.63E-09	1.85E-06	1.82E-03	2.16E-09	2.16E-09
F_3	M	1	1.29E+00	1.31E-01	4.04E+00	1.45E-01	**1.08E+00**	1.17E-05	2.51E-00	2.51E-00
F_4	M	0	3.8385-02	2.46E-02	**3.84E-08**	3.64E-08	9.42E-02	3.37E-02	4.35E-06	4.35E-06
F_5	M	0	5.38E-02	3.81E-02	**4.64E-04**	4.79E-04	3.37E+00	6.17E-01	4.92E-04	4.92E-04
F_6	M	0	3.47E-03	2.21E-01	**3.96E-06**	5.10E-06	2.87E+00	1.56E+00	7.87E-03	7.87E-03
F_7	M	0	4.261E-01	4.30E-01	**1.08E-05**	1.64E-05	1.37E+00	7.24E-01	1.16E-05	1.16E-05
F_8	U	0	2.62E-04	2.49E-04	4.81E-06	5.97E-05	1.58E-04	1.83E-03	**3.29E-06**	3.29E-06
F_9	U	0	4.76E+00	3.17E+00	1.35E+00	3.72E-02	**2.01E-03**	1.52E-03	5.94E-2	5.94E-2
F_{10}	U	0	2.82E-02	2.11E-02	9.22E+01	4.12E+01	1.76E-03	1.25E-04	**9.22E-04**	9.22E-04
F_{11}	M	0	7.05E-03	6.03E-03	4.65E-04	2.59E-04	6.06E-02	2.35E-02	**1.99E-04**	1.99E-04
F_{12}	U	0	5.06E-05	3.41E-05	7.65E-02	6.31E-02	7.67E-07	1.05E-07	**2.97E-07**	2.97E-07
F_{13}	M	0	3.29E-04	1.95E-04	**1.05E-07**	1.41E-07	5.68E-04	3.53E-02	1.35E-07	1.35E-07
F_{14}	U	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_{15}	M	0	2.94E-17	2.07-16	**3.14E-25**	1.06E-24	6.09E-10	1.19E-09	3.09E-21	3.09E-21
F_{16}	M	0	5.71E-01	3.06E-01	**8.19E-02**	1.63E-02	6.54E-01	1.31E-01	9.70E-02	9.70E-02

5.5.2 Performance of SVR-*e*ABO against Benchmarks on Standard Optimization Functions

Table 5.10

Comparison against Benchmarks on Standard Optimisation functions

			SVR-PSO		SVR-ABC		SVR-GA		SVR-CS		SVR-eABO	
Function	**C**	**GO**	**Mean**	**Stdv**	**Mean**	**Stdv**	**Mean**	**Stdv**	**Mean**	**Stdv**	**Mean**	**Stdv**
F_1	M	0	5.35E-04	3.22E-04	6.85E-07	5.38E-07	2.22E-02	1.33E-02	4.98E-04	4.71E-04	**8.31E-08**	2.52E-08
F_2	U	0	5.29E-05	5.27E-05	6.89E-07	8.11E-07	2.00E-03	1.62E-03	5.38E-06	7.41E-06	**2.16E-09**	4.52E-09
F_3	M	1	1.21E+00	7.86E-01	**1.48E-03**	1.81E-03	4.54E+00	1.05E+00	1.27E-02	3.82E-02	2.51E-00	2.30E-02
F_4	M	0	7.10E-02	2.89E-02	1.16E-02	4.07E-03	1.03E-01	3.74E-02	2.92E-02	2.60E-02	**4.35E-06**	2.90E-08
F_5	M	0	3.35E+00	1.12E+00	1.39E-01	1.04E-01	3.27E+00	6.28E-01	2.83E-01	3.92E-01	**4.92E-04**	5.07E-04
F_6	M	0	4.46E-01	2.66E-01	**5.03E-03**	3.86E-03	3.07E+00	1.86E+00	3.49E+00	4.48E+00	7.87E-03	5.35E-03
F_7	M	0	4.82E-01	4.11E-01	1.96E-03	1.74E-03	1.54E+00	7.00E-01	4.20E-02	2.07E-02	**1.16E-05**	1.93E-05
F_8	U	0	1.49E-03	2.01E-03	2.63E-04	3.32E-04	2.47E-03	3.24E-03	3.06E-03	3.69E-03	**3.29E-06**	2.17E-05
F_9	U	0	8.92E+00	6.79E+00	**5.92E-02**	7.57E-02	1.96E+01	1.42E+01	7.33E-01	5.92E-01	5.94E-02	3.91E-02
F_{10}	U	0	9.50E+01	5.46E+01	2.78E-02	2.35E-02	1.81E+01	1.30E+01	1.86E+01	4.28E+01	**9.22E-04**	6.83E-03
F_{11}	M	0	4.05E-02	1.52E-02	8.09E-03	3.90E-03	6.40E-02	2.46E-02	5.39E-02	2.99E-02	**1.99E-04**	2.73E-04
F_{12}	U	0	7.70E-04	6.72E-04	8.17E-07	6.66E-07	1.01E-02	9.81E-03	9.03E-04	7.92E-03	**2.97E-07**	3.06E-07
F_{13}	M	0	2.98E-03	1.96E-03	5.73E-05	4.20E-05	6.79E-02	4.08E-02	4.91E-02	4.28E-04	**1.35E-07**	1.62E-07
F_{14}	U	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_{15}	M	0	5.54E-14	1.07E-13	7.19E-18	1.01E-17	5.86E-10	1.12E-09	6.29E-09	1.35E-11	**3.09E-21**	2.75E-21
F_{16}	M	0	3.96E-01	1.95E-01	0.00E+00	0.00E+00	6.17E-01	1.71E-01	4.07E-01	1.91E+00	**9.70E-02**	7.71E-02

5.6 Comparison of CPU Time Based on Standard Optimisation Functions

This section presents the performance of all the developed algorithms in this study viz SVR-*pop*ABO, SVR-*explr*ABO, SVR-*explt*ABO, and the final SVR-*e*ABO algorithms in terms of CPU execution time. Likewise, the performance of the final SVR-*e*ABO algorithm has been compared with several benchmarks on the same CPU time metrics. The following sub-sections presented the detailed analysis of the obtained results.

5.6.1 CPU Execution Time of Standard Optimization Functions: Proposed Algorithms

The data in table 5.11 reveals a distinct comparative performance assessment of the four algorithms under evaluation: SVR-*pop*ABO, SVR-*explr*ABO, SVR-*explt*ABO, and SVR-*e*ABO. A careful examination of the execution times demonstrates that the SVR-*e*ABO algorithm consistently exhibits the lengthiest CPU processing times across the majority of the benchmark functions. Conversely, the SVR-*pop*ABO and SVR-*explr*ABO algorithms emerge as the most efficient, displaying the shortest execution times for the vast preponderance of the tested functions.

Further scrutiny of the performance trends indicates that Sphere, SumSquares, Schaffer2, and Zakharov, consistently exhibit relatively lower execution times across all four algorithms in comparison to the other functions evaluated. Conversely, the Weierstrass function stands out as an outlier, with significantly elevated execution times, particularly for the SVR-*e*ABO algorithm. Additionally, the Whitley, Griewank, Schwefel, and Dixonprice functions also demonstrate comparatively higher execution times irrespective of the algorithm employed.

The comparative analysis suggests that the SVR-*pop*ABO and SVR-*explr*ABO algorithms possess a distinct advantage in terms of computational efficiency, as evidenced by their consistently expeditious execution times. Conversely, the SVR-*e*ABO algorithm, while potentially offering enhanced performance in other facets, appears to be the most computationally intensive of the four, exhibiting the lengthiest CPU processing times. The SVR-*explt*ABO algorithm's performance falls within the intermediate range, striking a balance between execution time and potentially other relevant performance metrics.

Table 5.11

Comparison of developed algorithms on Standard Optimisation functions

Functions	**SVR-*pop*ABO**	**SVR-*explr*ABO**	**SVR-*explt*ABO**	**SVR-*e*ABO**
Sphere	**2.99E+00**	3.54E+00	3.15E+00	5.82E+00
SumSquares	2.76E+00	2.39E+00	**2.28E+00**	3.90E+00
Whitley	**4.38E+00**	4.62E+00	4.57E+00	7.03E+00
Griewank	**2.96E+00**	3.81E+00	3.77E+00	6.99E+00
Ackley	**3.35E+00**	4.07E+00	3.82E+00	5.03E+00
Pinter	**3.24E+00**	3.43E+00	3.39E+00	5.17E+00
Rastrigin	**2.11E+00**	3.71E+00	3.61E+00	4.91E+00
Schaffer2	**2.13E+00**	2.51E+00	2.48E+00	3.97E+00
Rosenbrock	**3.16E+00**	4.07E+00	3.97E+00	5.33E+00
Schwefel	**4.17E+00**	4.92E+00	4.63E+00	5.62E+00
Alpine	3.59E+00	3.64E+00	**3.48E+00**	4.41E+00
Dixonprice	2.98E+00	3.01E+00	**2.79E+00**	4.88E+00
Zakharov	**2.38E+00**	2.69E+00	2.42E+00	3.49E+00
Powell	**2.27E+00**	2.74E+00	2.61E+00	4.08E+00
Infinity	3.36E+00	3.42E+00	**3.28E+00**	4.25E+00
Weierstrass	**8.32E+00**	1.02E+01	9.07E+00	1.88E+02

5.6.2 CPU Execution Time of Standard Optimization Functions: Proposed Algorithms vs. Benchmarks

Table 5.12 presents the execution time (CPU time) performance of the developed algorithms - SVR-PSO, SVR-ABC, SVR-GA, SVR-CS, and SVR-*e*ABO - across a variety of benchmark functions. A thorough analysis of the data reveals several key insights. Firstly, the SVR-PSO algorithm consistently exhibits the shortest execution times across the majority of the benchmark functions, demonstrating its superior computational efficiency. This is particularly evident in functions such as Sphere, SumSquares, Schaffer2, Powell, and Zakharov, where the SVR-PSO algorithm significantly outperforms the other algorithms.

In contrast, the SVR-*e*ABO algorithm appears to be the most computationally intensive, displaying the longest execution times for almost all of the benchmark functions. This is particularly noticeable in the Weierstrass function, where the SVR-*e*ABO algorithm exhibits an exceptionally high CPU time of 2.100E+01, significantly higher than the other algorithms.

The SVR-ABC and SVR-GA algorithms occupy an intermediate performance range, generally exhibiting longer execution times than the SVR-PSO algorithm but shorter times than the SVR-*e*ABO algorithm. The SVR-CS algorithm's performance falls within a similar range, with some functions, such as Sphere and SumSquares, showing relatively efficient execution times, while others, like Weierstrass and Alpine, demonstrate comparatively longer CPU processing times.

It is also noteworthy that certain benchmark functions, such as Sphere, SumSquares, and Zakharov, consistently exhibit lower execution times across all five algorithms, suggesting that these functions may be relatively less computationally demanding. Conversely, functions like Weierstrass, Schwefel, and Alpine appear to be more computationally intensive, with significantly higher execution times observed across the algorithms.

Table 5.12

Comparison against Benchmarks on Standard Optimisation functions

Functions	**SVR-PSO**	**SVR-ABC**	**SVR-GA**	**SVR-CS**	**SVR-*e*ABO**
Sphere	2.72E+00	4.48E+00	6.92E+00	3.30E+00	7.04E+00
SumSquares	2.78E+00	4.46E+00	7.03E+00	3.81+00	7.98E+00
Whitley	4.29E+00	5.65E+00	8.56E+00	5.20E+00	10.04E+00
Griewank	2.96E+00	4.62E+00	7.10E+00	5.11E+00	8.93E+00
Ackley	3.72E+00	5.51E+00	7.92E+00	6.18E+00	10.17E+00
Pinter	3.49E+00	5.00E+00	8.05E+00	4.72E+00	8.79E+00
Rastrigin	2.83E+00	4.17E+00	6.95E+00	7.31E+00	9.07E+00
Schaffer2	2.73E+00	4.60E+00	7.25E+00	4.15E+00	8.59E+00
Rosenbrock	3.00E+00	4.54E+00	7.81E+00	6.43E+00	9.37E+00
Schwefel	5.07E+00	6.63E+00	9.83E+00	5.99E+00	12.77E+00
Alpine	3.65E+00	5.59E+00	8.75E+00	8.35E+00	11.07E+00
Dixonprice	2.92E+00	4.73E+00	7.37E+00	5.33E+00	9.94E+00
Zakharov	2.77E+00	4.38E+00	6.91E+00	7.630E+00	10.83E+00
Powell	2.38E+00	3.71E+00	6.51E+00	5.74E+00	9.57E+00
Infinity	3.27E+00	5.03E+00	7.44E+00	5.53E+00	9.35E+00
Weierstrass	8.46E+00	1.05E+01	1.41E+01	1.85E+01	2.10+01

5.7 Summary

This chapter presents a comprehensive evaluation of the performance of several enhanced variants of the SVR-ABO algorithm for electricity forecasting tasks. The algorithms under investigation include SVR-ABO, SVR-*pop*ABO, SVR-*explr*ABO, SVR-*explt*ABO, and SVR-*e*ABO, which were tested on four real-world datasets: the

Household dataset, the Panama electricity consumption dataset, the Appliances electricity consumption dataset, and the Turkey electricity consumption dataset. The study compared the performance of the developed SVR-ABO algorithms against SVR, SVR-PSO, SVR-ABC, SVR-GA, and SVR-CS as benchmarks. Additionally, the algorithms were evaluated using standard optimization functions to provide a broader assessment of their capabilities.

The results obtained from this evaluation demonstrate that the enhancements made to the original ABO algorithm have significantly contributed to improving its performance as an optimization algorithm for SVR. The enhanced versions, such as SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO, consistently outperformed the classical benchmarks in terms of finding optimal parameters for the SVR model, leading to better electricity forecasting accuracy across the tested datasets. Specifically, the SVR-*pop*ABO and SVR-*explr*ABO algorithms exhibited the most efficient computational performance, as evidenced by their shorter execution times compared to the other algorithms. In contrast, the SVR-*e*ABO algorithm, while potentially offering enhanced performance in other aspects, was found to be the most computationally intensive of the group.

Overall, this chapter provides valuable insights into the comparative performance of the developed variants of SVR-ABO algorithms and their ability to optimize SVR models for improved electricity forecasting accuracy. The findings demonstrate the effectiveness of the enhancements made to the original SVR-ABO algorithm and its potential for practical applications in the energy forecasting domain.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATION

In this concluding chapter, the key findings of the research were encapsulated reinforcing the significance of the enhanced algorithms developed throughout the thesis. The chapter reflects on the contributions made to both knowledge and practical applications, emphasizing how the advancements in the African Buffalo Optimization (ABO) algorithm have enriched the field of optimization and machine learning. The discussion extends to the practical implications of the work, illustrating the potential for real-world applications. Furthermore, the chapter outlined recommendations for future research endeavours, suggesting avenues for further exploration and enhancement of the algorithms. This chapter serves as a comprehensive wrap-up, summarizing the research journey and its impact while paving the way for continued innovation in the domain.

6.1 Conclusion

Forecasting of electricity has witnessed major changes in past decades. Beginning from conventional statistical techniques to Computational Intelligence (CI) based approaches which attracted attention of both academia and practitioners' communities. This is due to the peculiar nature of the electricity as it cannot be stored in large quantities for future consumption taking into cognizance of the importance of electric energy in present day economy as highlighted in chapter one. With respect to that matter, vast studies regarding electricity load forecasting have been a promising, and is presently an active area of research. This has been ascertained as demonstrated in Chapter Two of this report. It is from the studied literature that the discovered existing gaps in the forecasting technique that led to the developed SVR-*e*ABO algorithm through various stages. This has been proven to be essential as the forecasting accuracy is of paramount importance.

Through using various real datasets of electricity load consumption, the SVR and ABO algorithms were hybridised, for the purpose of optimising the SVR hyper-parameters. This was meant to achieve the objective one of this study as highlighted in section 1.7. In order to address the highlighted problems of ABO algorithm (see Chapter One, section 1.5), enhancements introduced to the standard ABO algorithm were proven to be of significance in respect of the problem under study.

A novel population generation function based on the Tent-map function was introduced to replace the existing population generation method of the algorithm. The objective of this enhancement was to generate a population of buffaloes with maximum diversity, thereby facilitating improved solution generation. The outcomes of the experimentation reveal a notable increase in the convergence speed across different datasets. The enhanced algorithm demonstrates a faster convergence rate, implying that it is capable of reaching optimal solutions more efficiently. This improvement in convergence speed can be attributed to the enhanced diversity achieved through the new population generation function.

The exploration function of the ABO algorithm was subsequently enhanced through the application of the McCulloch algorithm, utilizing a Lévy mutation approach. This enhancement aimed to generate random values with varying magnitudes and sporadic occurrence of large values. By doing so, it facilitated the computation of the next position of the buffalo within the search space, thereby mitigating the aimless search characteristic of the ABO algorithm. This improvement effectively addresses the potential issues of over-fitting and under-fitting. Consequently, the second objective of this research has been successfully accomplished.

Furthermore, the exploitation behaviour of the ABO algorithm has been enhanced through the utilisation of the proposed function in section 3.3.5, specifically incorporating the Tent-map-based approach. The integration of the Tent-map-based exploitation function ensures a more comprehensive search of the solution space, allowing the algorithm to overcome the limitations associated with local optima and improve its overall performance.

In conclusion, the empirical results obtained from the implementation of the enhancements in the ABO algorithm, as presented in the various stages discussed above, demonstrate the improved forecasting capabilities of the developed hybrid SVR-*e*ABO algorithm. Notably, the combined implementation of SVR-*pop*ABO, SVR-*explr*ABO, and SVR-*explt*ABO yields significantly improved forecasting results. By incorporating these enhancements, the new developed hybrid SVR-*e*ABO algorithm is able to overcome the issue of premature convergence, increased speed of convergence and effective exploration in search space, thereby leading to more accurate and reliable forecasting. Consequently, the fourth objective of this study has been successfully accomplished.

6.2 Contribution

Several contributions have been made in the course achieving SVR-*e*ABO, which can be broadly classified into two categories: knowledge-based contributions and practical-based contributions. Each of these categories can be further subdivided into specific subcategories, as delineated in the subsequent sections.

6.2.1 Knowledge Contribution

This study contributes the following contributions:

i. SVR-ABO algorithm that can automatically optimised SVR parameters (i.e., cost error (C), tube size (ε) and kernel parameter (γ). This is defined as objective one.

ii. SVR-*pop*ABO algorithm that offers maximum diversity of population generation. The improvement in searching capability thereby increases the convergence speed of the developed SVR-*e*ABO algorithm. This is defined as objective two.

iii. SVR-*explr*ABO (presented as objective three) algorithm that addresses local minima problem by enhancing the exploration ability of ABO algorithm by preventing the buffaloes from aimless searching. This helps the algorithm to have maximum potential in finding optimal solution.

iv. The SVR-*explt*ABO algorithm that also prevents the ABO algorithm from falling into local optima. This is achieved through enhancing the exploitation ability of the ABO algorithm. This is defined as objective four.

v. The SVR-*e*ABO algorithm that combines all improvements made in ABO to escape from premature convergence and increase convergence speed. This is defined as fifth objective.

6.2.2 Practical Contribution

As all the algorithms developed in this study were tested on real-world datasets, the results obtained shows that the algorithms can make good electricity consumption forecast. Hence, this study has practical contributions as follows:

i. Improved resource planning: Accurate forecasts help utility companies plan power generation and transmission capacity effectively, ensuring sufficient resources are available to meet demand and minimizing the risk of shortages or overcapacity.

ii. Cost optimisation: Accurate demand forecasts enable companies to optimise operations and resource allocation, including power generation schedules, electricity purchases, and energy storage management. This leads to cost savings and efficient resource utilization.

iii. Demand response management: Accurate electricity forecasts could allow utility companies to anticipate peak demand periods and encourage consumers to reduce electricity usage during those times. This helps balance the grid, reduce strain, and prevent blackouts or disruptions.

iv. Energy efficiency initiatives: Accurate forecasts provide insights into consumption patterns and highlight areas for energy efficiency improvements. This information guides energy conservation initiatives, promotes sustainable practices, and reduces overall energy consumption.

Overall, a model built for electricity forecasting can contribute to improved operational efficiency, cost savings, grid stability, renewable energy integration, and promoting energy conservation efforts.

6.3 Recommendations for Future Works

Based on the results and discussion presented in Chapter Five (5), the proposed SVR-*e*ABO hybrid algorithm has been proven to be more superior based on metrics as compared to other forecasting algorithms. This indicates that the SVR-*e*ABO possess significant capability on the problem of interest. However, there is always other areas that require further improvement in order to improve upon potential yet to be discovered limitations.

To begin with, classical ABO algorithm has a great speed of convergence as mentioned in Odili & Noraziah, (2018). However, upon the execution of the proposed improvements it has been noticed that the speed of convergence reduced to some extent. This proves that achieving both speed of convergence and avoiding local optima entrapment is a challenging task. However, it is an interesting challenge that can be considered as future work to improve the speed of the algorithm without sacrificing of its efficiency.

In addition, the algorithm performance was tested on a single computer system. It will be interesting to measure the performance on grid computing architecture. By so doing, the jobs between the buffaloes in ABO algorithm could be distributed to various computing resources. This may lead to faster and more efficient computation.

REFERENCES

Abbaszadehpeivasti, H., de Klerk, E., & Zamani, M. (2022). The exact worst-case convergence rate of the gradient method with fixed step lengths for L-smooth functions. *Optimization Letters, 16*(6), 1649–1661. https://doi.org/10.1007/s11590-021-01821-1

Agga, A., Abbou, A., & Labbadi, M. (2022). Day-Ahead Photovoltaic Power Production Forecasting Following Traditional and Hierarchical Approach. In *Lecture Notes in Networks and Systems: Vol. 455 LNNS*. https://doi.org/10.1007/978-3-031-02447-4_18

Agrawal, R. K., Muchahary, F., & Tripathi, M. M. (2018). Long term load forecasting with hourly predictions based on long-short-term-memory networks. *2018 IEEE Texas Power and Energy Conference, TPEC 2018*, *2018-Febru*, 1–6. https://doi.org/10.1109/TPEC.2018.8312088

Algaphari, G. H. (2023). *An Efficient African Buffalo Optimization Algorithm for Traveling Salesman Problem Using Fuzzy Matrix*. 0–11.

Almonacid, B. (2017). Simulation of a Dynamic Prey-Predator Spatial Model Based on Cellular Automata Using the Behavior of the Metaheuristic African Buffalo Optimization. *IWINAC: International Work-Conference on the Interplay Between Natural and Artificial Computation*, *10337*, 170–180. https://doi.org/10.1007/978-3-319-59740-9

Almonacid, B., Reyes-Hagemann, J., Campos-Nazer, J., & Ramos-Aguilar, J. (2017). Selecting a biodiversity conservation area with a limited budget using the binary African buffalo optimisation algorithm. *IET Software*, *12*(2), 96–111. https://doi.org/10.1049/iet-sen.2016.0327

Al-Musaylh, M. S., Deo, R. C., Adamowski, J. F., & Li, Y. (2018). Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia. *Advanced Engineering Informatics, 35*(April 2017), 1–16. https://doi.org/10.1016/j.aei.2017.11.002

Al-Qahtani, F. H., & Crone, S. F. (2013). Multivariate k-nearest neighbour regression for time series data - A novel algorithm for forecasting UK electricity demand. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2013.6706742

Alweshah, M., Rababa, L., Ryalat, M. H., Al Momani, A., & Ababneh, M. F. (2022). African buffalo algorithm: Training the probabilistic neural network to solve classification problems. *Journal of King Saud University - Computer and Information Sciences*, *34*(5), 1808–1818. https://doi.org/10.1016/j.jksuci.2020.07.004

Arif, M., Chen, J., Wang, G., & Rauf, H. T. (2022). Cognitive population initialization for swarm intelligence and evolutionary computing. *Journal of Ambient Intelligence and Humanized Computing*, *13*(12), 5847–5860. https://doi.org/10.1007/s12652-021-03271-0

Ashraf, A., Pervaiz, S., Haider Bangyal, W., Nisar, K., Ag. Ibrahim, A. A., Rodrigues, J. J. P. C., & Rawat, D. B. (2021). Studying the impact of initialization for population-based algorithms with low-discrepancy sequences. *Applied Sciences (Switzerland)*, *11*(17), 1–41. https://doi.org/10.3390/app11178190

Avatefipour, O., & Nafisian, A. (2018). A novel electric load consumption prediction and feature selection model based on modified clonal selection algorithm. *Journal of Intelligent and Fuzzy Systems*, *34*(4), 2261–2272. https://doi.org/10.3233/JIFS-171292

Azad, M. K., Uddin, S., & Takruri, M. (2018). Support Vector Regression Based Electricity Peak Load Forecasting. *11th International Symposium on Mechatronics and Its Applications, ISMA 2018*, *2018-Janua*, 1–5. https://doi.org/10.1109/ISMA.2018.8330143

B, A. K., Alhwaiti, Y., Leider, A., & Tappert, C. C. (2019). A Two-Fold Machine Learning Approach for Efficient Day-Ahead Load Prediction at Hourly Granularity for NYC. *Advances in Information and Communication. FICC 2019*, *70*, 84–97. https://doi.org/10.1007/978-3-030-12385-7

Bandyopadhyay, S., Hazra, J., & Kalyanaraman, S. (2018). A machine learning based heating and cooling load forecasting approach for DHC networks. *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 1–5. https://doi.org/10.1109/ISGT.2018.8403331

Barman, M., & Dev Choudhury, N. B. (2018). Hybrid GOA-SVR technique for short term load forecasting during periods with substantial weather changes in North-East India. *Procedia Computer Science*, *143*, 124–132. https://doi.org/10.1016/j.procs.2018.10.360

Barman, M., Dev Choudhury, N. B., & Sutradhar, S. (2018). A regional hybrid GOA-SVM model based on similar day approach for short-term load forecasting in Assam, India. *Energy*, *145*, 710–720. https://doi.org/10.1016/j.energy.2017.12.156

Barnwal, S. K., Prakash, A., & Yadav, D. K. (2023). Improved African Buffalo Optimization-Based Energy Efficient Clustering Wireless Sensor Networks using Metaheuristic Routing Technique. *Wireless Personal Communications*, *130*(3), 1575–1596. https://doi.org/10.1007/s11277-023-10345-z

Bashath, S., Ismail, A. R., Alwan, A. A., & Hussin, A. A. A. (2022). An improved particle swarm optimization based on lévy flight and simulated annealing for high dimensional optimization problem. *International Journal of Advances in Intelligent Informatics*, *8*(1), 115–134. https://doi.org/10.26555/ijain.v8i1.818

Bergsteinsson, H. G., Sørensen, M. L., Møller, J. K., & Madsen, H. (2023). Heat load forecasting using adaptive spatial hierarchies. *Applied Energy*, *350*, 121676. https://doi.org/https://doi.org/10.1016/j.apenergy.2023.121676

Bing, Q., Qu, D., Chen, X., Pan, F., & Wei, J. (2018). Short-Term Traffic Flow Forecasting Method Based on LSSVM Model Optimized by GA-PSO Hybrid Algorithm. *Discrete Dynamics in Nature and Society*, *2018*, 10. https://doi.org/10.1155/2018/3093596

Boser, E., Vapnik, N., & Guyon, I. M. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152. https://doi.org/10.1145/130385.130401

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, *11*(7). https://doi.org/10.3390/en11071636

Cabello-López, T., Carranza-García, M., Riquelme, J. C., & García-Gutiérrez, J. (2023). Forecasting solar energy production in Spain: A comparison of univariate and multivariate models at the national level. *Applied Energy*, *350*, 121645. https://doi.org/https://doi.org/10.1016/j.apenergy.2023.121645

Candanedo, L. (2017). *Appliances energy prediction*. UCI Machine Learning Repository. https://doi.org/https://doi.org/10.24432/C5VC8G

Candanedo, L. M., Feldheim, V., & Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, *140*, 81–97. https://doi.org/10.1016/j.enbuild.2017.01.083

Caraka, R. E., Bakar, S. A., Pardamean, B., & Budiarto, A. (2018). Hybrid support vector regression in electric load during national holiday season. *Proceedings - 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech 2017*, *2018-Janua*, 1–6. https://doi.org/10.1109/INNOCIT.2017.8319127

Chae, Y. T., Horesh, R., Hwang, Y., & Lee, Y. M. (2016). Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. *Energy and Buildings*, *111*, 184–194. https://doi.org/10.1016/j.enbuild.2015.11.045

Chang, J.-R., Liu, Z.-Q., & Yu, P.-Y. (2019). Forecasting for the total electricity consumption of Taiwan by fuzzy time series model. *Smart Innovation, Systems and Technologies*, *110*, 220–225. https://doi.org/10.1007/978-3-030-03748-2_27

Che, J. X., Yang, Y. L., Li, L., Li, Y. Y., & Zhu, S. L. (2017). A Modified Support Vector Regression: Integrated Selection of Training Subset and Model. *Journal of Applied Soft Computing*, *53*, 308–322. https://doi.org/10.1016/j.asoc.2016.12.053

Chen, Y. H., Hong, W. C., Shen, W., & Huang, N. N. (2016). Electric load forecasting based on a least squares support vector machine with fuzzy time series and global harmony search algorithm. *Energies*, *9*(2), 1–13. https://doi.org/10.3390/en9020070

Cheung, C. M., Kannan, R., & Prasanna, V. K. (2018). Temporal Ensemble Learning of Univariate Methods for Short Term Load Forecasting. *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 1–5. https://doi.org/10.1109/ISGT.2018.8403378

Chitsaz, H., Shaker, H., Zareipour, H., Wood, D., & Amjady, N. (2015). Short-term electricity load forecasting of buildings in microgrids. *Energy and Buildings*, *99*, 50–60. https://doi.org/10.1016/j.enbuild.2015.04.011

Chou, J. S., Ngo, N. T., & Chong, W. K. (2017). The use of artificial intelligence combiners for modeling steel pitting risk and corrosion rate. *Engineering Applications of Artificial Intelligence*, *65*, 471–483. https://doi.org/10.1016/j.engappai.2016.09.008

Chou, J. S., & Pham, A. D. (2017). Nature-inspired metaheuristic optimization in least squares support vector regression for obtaining bridge scour information. *Information Sciences*, *399*, 64–80. https://doi.org/10.1016/j.ins.2017.02.051

Chou, J.-S., & Truong, T. T. H. (2019). Sliding-window metaheuristic optimization-based forecast system for foreign exchange analysis. *Soft Computing*. https://doi.org/10.1007/s00500-019-03863-1

Chow, H. F. (2021). Short-term electricity grid maximum demand forecasting with the ARIMAX-SVR machine learning hybrid model. *HKIE Transactions Hong Kong Institution of Engineers*, *28*(1), 22–30. https://doi.org/10.33430/V28N1THIE-2020-0005

Chuang, C. C., Su, S. F., Jeng, J. T., & Hsiao, C. C. (2002). Robust support vector regression networks for function approximation with outliers. *IEEE Transactions on Neural Networks*, *13*(6), 1322–1330. https://doi.org/10.1109/TNN.2002.804227

Cinar, Y. G., Mirisaee, H., Goswami, P., Gaussier, E., & Aït-Bachir, A. (2018). Period-aware content attention RNNs for time series forecasting with missing values. *Neurocomputing*, *312*, 177–186. https://doi.org/10.1016/j.neucom.2018.05.090

Conejo, A. J., Plazas, M. A., Espínola, R., & Molina, A. B. (2005). Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. *IEEE Transactions on Power Systems*, *20*(2), 1035–1042. https://doi.org/10.1109/TPWRS.2005.846054

Corinna, C., & Vladimir, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Dash, S. S., Nayak, S. K., & Mishra, D. (2021). ABC Versus PSO: A Comparative Study and Analysis on Optimization Aptitude. In S. Das & M. N. Mohanty (Eds.), *Advances in Intelligent Computing and Communication* (pp. 527–544). Springer Singapore.

Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, *74*(July 2016), 902–924. https://doi.org/10.1016/j.rser.2017.02.085

Deng, W., Chen, H., & Li, H. (2014). A novel hybrid intelligence algorithm for solving combinatorial optimization problems. *Journal of Computing Science and Engineering*, *8*(4), 199–206. https://doi.org/10.5626/JCSE.2014.8.4.199

Dieudonné, N. T., Armel, T. K. F., Hermann, D. T., Vidal, A. K. C., & René, T. (2023). Optimization of Short-Term Forecast of Electric Power Demand in the city of Yaoundé-Cameroon by a hybrid model based on the combination of neural networks and econometric methods from a designed energy optimization algorithm. *Technological Forecasting and Social Change*, *187*(November 2022). https://doi.org/10.1016/j.techfore.2022.122212

Dong, Y., Zhang, Z., & Hong, W. C. W.-C. (2018). A hybrid seasonal mechanism with a chaotic cuckoo search algorithm with a support vector regression model for electric load forecasting. *Energies*, *11*(4). https://doi.org/10.3390/en11041009

Dung, N. T., Phuong, N. T., & Vinh, N. H. (2021). Impact of Covid-19 and SLP-SVR algorithms on short-term load forecast, case study: EVNHCMC. *Proceedings of 2021 International Conference on System Science and Engineering, ICSSE 2021*, 7–12. https://doi.org/10.1109/ICSSE52999.2021.9538425

Eapen, R. R., & Simon, S. P. (2018). Performance Analysis of Combined Similar Day and Day Ahead Short-Term Electrical Load Forecasting using Sequential Hybrid Neural

Networks. *IETE Journal of Research*, *2063*(May), 1–11. https://doi.org/10.1080/03772063.2017.1417749

Eberhart, R., & Kennedy, J. (2016). A New Optimizer Using Particle Swarm Theory. *Annals of Surgical Oncology*, *23*(10), 3162–3167. https://doi.org/10.1245/s10434-016-5403-0

El-Ashmawi, W. H. (2018). An Improved African Buffalo Optimization Algorithm for Collaborative Team Formation in Social Network. *International Journal of Information Technology and Computer Science*, *10*(5), 16–29. https://doi.org/10.5815/ijitcs.2018.05.02

Ellwanger, R., & Snudden, S. (2023). Forecasts of the real price of oil revisited: Do they beat the random walk? *Journal of Banking & Finance*, *154*, 106962. https://doi.org/https://doi.org/10.1016/j.jbankfin.2023.106962

Fan, G. F., Yu, M., Dong, S. Q., Yeh, Y. H., & Hong, W. C. (2021). Forecasting short-term electricity load using hybrid support vector regression with grey catastrophe and random forest modeling. *Utilities Policy*, *73*. https://doi.org/10.1016/j.jup.2021.101294

Farah, A., & Belazi, A. (2018). A novel chaotic Jaya algorithm for unconstrained numerical optimization. *Nonlinear Dynamics*, *93*(3), 1451–1480. https://doi.org/10.1007/s11071-018-4271-5

Fu, Y., Li, Z., Zhang, H., & Xu, P. (2015). Using Support Vector Machine to Predict Next Day Electricity Load of Public Buildings with Sub-metering Devices. *Procedia Engineering*, *121*, 1016–1022. https://doi.org/10.1016/j.proeng.2015.09.097

Gasparin, A., Lukovic, S., & Alippi, C. (2022a). Deep learning for time series forecasting: The electric load case. *CAAI Transactions on Intelligence Technology*, *7*(1), 1–25. https://doi.org/10.1049/cit2.12060

Gasparin, A., Lukovic, S., & Alippi, C. (2022b). Deep learning for time series forecasting: The electric load case. *CAAI Transactions on Intelligence Technology*, *7*(1), 1–25. https://doi.org/10.1049/cit2.12060

Gerolimetto, M., & Bisaglia, L. (2014). Testing for (non)linearity in economic time series : a Montecarlo comparison. *Quaderni Di Statistica*, *16*(16), 5–32.

Ghosh, T. (2022). An Industrial Application of Cellular Manufacturing Using African Buffalo Optimization. In *Lecture Notes in Electrical Engineering: Vol. 880 LNEE*. https://doi.org/10.1007/978-981-19-0572-8_93

Gopal Dhal, K., Quraishi, I., & Das, S. (2015). A Chaotic Lévy flight Approach in Bat and Firefly Algorithm for Gray level image Enhancement. *International Journal of Image, Graphics and Signal Processing*, *7*(7), 69–76. https://doi.org/10.5815/ijigsp.2015.07.08

Gündüz, M., Kiran, M. S., & Özceylan, E. (2015). A hierarchic approach based on swarm intelligence to solve the traveling salesman problem. *Turkish Journal of Electrical Engineering and Computer Sciences*, *23*(1), 103–117. https://doi.org/10.3906/elk-1210-147

Guo, Y. M., He, P., Wang, X. T., Zheng, Y. F., Liu, C., & Cai, X. B. (2015). Fast prediction with sparse multikernel LS-SVR using multiple relevant time series and its application in avionics system. *Mathematical Problems in Engineering*, *2015*. https://doi.org/10.1155/2015/460514

Halaš, P., Lóderer, M., & Rozinajová, V. (2017). Prediction of electricity consumption using biologically inspired algorithms. *2017 IEEE 14th International Scientific Conference on Informatics*, *2018-Janua*, 98–103. https://doi.org/10.1109/INFORMATICS.2017.8327229

Hall, D., & Nguyen, T. A. (2017). Electricity Liberalisation in Developing Countries. *Progress in Development Studies*, *17*(2), 99–115. https://doi.org/10.1177/1464993416688824

Hamzaçebi, C. (2008). Improving artificial neural networks' performance in seasonal time series forecasting. *Information Sciences*, *178*(23), 4550–4559. https://doi.org/10.1016/j.ins.2008.07.024

Hebrail, Georges and Berard, A. (2012). *Individual household electric power consumption*. UCI Machine Learning Repository. https://doi.org/https://doi.org/10.24432/C58K54

Hermias, J. P., Teknomo, K., & Monje, J. C. N. (2018). Short-term stochastic load forecasting using autoregressive integrated moving average models and Hidden Markov Model. *2017 International Conference on Information and Communication Technologies, ICICT 2017*, *2017-Decem*, 131–137. https://doi.org/10.1109/ICICT.2017.8320177

Hu, Z., Bao, Y., & Xiong, T. (2014). Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression. *Applied Soft Computing Journal*, *25*, 15–25. https://doi.org/10.1016/j.asoc.2014.09.007

Huang, B., Wu, D., Lai, C. S., Cun, X., Yuan, H., Xu, F., Lai, L. L., & Tsang, K.-F. (2018). Load Forecasting based on Deep Long Short-term Memory with Consideration of Costing Correlated Factor. *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018*, 496–501. https://doi.org/10.1109/INDIN.2018.8472040

Huang, C. J., & Kuo, P. H. (2018). A deep cnn-lstm model for particulate matter (Pm2.5) forecasting in smart cities. *Sensors (Switzerland)*, *18*(7). https://doi.org/10.3390/s18072220

Humeau, S., Wijaya, T. K., Vasirani, M., & Aberer, K. (2013). Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. *2013 Sustainable Internet and ICT for Sustainability, SustainIT 2013*. https://doi.org/10.1109/SustainIT.2013.6685208

Huo, J., Shi, T., & Chang, J. (2017). Comparison of Random Forest and SVM for electrical short-Term load forecast with different data sources. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 1077–1080. https://doi.org/10.1109/ICSESS.2016.7883252

Hussain, A., Rahman, M., & Memon, J. A. (2016). Forecasting electricity consumption in Pakistan: The way forward. *Energy Policy*, *90*, 73–80. https://doi.org/10.1016/j.enpol.2015.11.028

Igiri, C. P., Singh, Y., & Bhargava, D. (2019a). An improved African buffalo optimization algorithm using chaotic map and chaotic - levy flight. *International Journal of Engineering & Technology*, *7*(4), 4570–4576.

Igiri, C. P., Singh, Y., & Bhargava, D. (2019b). An Improved African Buffalo Optimization Algorithm using Chaotic Map and Chaotic-Levy Flight. *International Journal of Engineering & Technology*, *7*(4), 4570–4576.

Igiri, C. P., Singh, Y., Bhargava, D., & Shikaa, S. (2020). Improved African Buffalo Optimisation algorithm for Petroleum Product Supply Chain Management. *International Journal of Grid and Utility Computing*, *11*(6), 769–779. https://doi.org/10.1504/IJGUC.2020.110905

Igiri, C. P., Singh, Y., & Poonia, R. C. (2019). An improved chaotic-based African buffalo optimisation algorithm. *International Journal of Innovative Computing and Applications*, *10*(3–4), 147–153. https://doi.org/10.1504/IJICA.2019.103375

Iliya, S., Goodyer, E., Gow, J., Shell, J., & Gongora, M. (2015). Application of artificial neural network and support vector regression in cognitive radio networks for RF power prediction using compact differential evolution algorithm. *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015*, 55–66. https://doi.org/10.15439/2015F14

Ingle, K. K., & Jatoth, R. K. (2023). Non-linear channel equalization using modified grasshopper optimization algorithm. *Applied Soft Computing*, 110091. https://doi.org/https://doi.org/10.1016/j.asoc.2023.110091

Invernizzi, M., & Parrinello, M. (2022). Exploration vs Convergence Speed in Adaptive-Bias Enhanced Sampling. *Journal of Chemical Theory and Computation*, *18*(6), 3988–3996. https://doi.org/10.1021/acs.jctc.2c00152

Ismail, A. R., Aziz, N. A., Ralib, A. M., Abidin, N. Z., & Bashath, S. S. (2021). A particle swarm optimization levy flight algorithm for imputation of missing creatinine dataset. *International Journal of Advances in Intelligent Informatics*, *7*(2), 225–236. https://doi.org/10.26555/ijain.v7i2.677

Jaddi, N. S., Abdullah, S., & Malek, M. A. (2017). Master-leader-slave cuckoo search with parameter control for ANN optimization and its real-world application to water quality prediction. *PLoS ONE*, *12*(1), 1–19. https://doi.org/10.1371/journal.pone.0170372

Jalalifar, S., Masoudi, M., Abbassi, R., Garaniya, V., Ghiji, M., & Salehi, F. (2019). A hybrid SVR-PSO model to predict a CFD-based optimised bubbling fluidised bed pyrolysis reactor. *Energy*, *191*, 116414. https://doi.org/10.1016/j.energy.2019.116414

Jamil, M., & Yang, X. S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, *4*(2), 150–194. https://doi.org/10.1504/IJMMNO.2013.055204

Jia, H. (2015). A novel hybrid optimization algorithm and its application in solving complex problem. *International Journal of Hybrid Information Technology*, *8*(2), 1–10.

Jiang, F., Peng, Z., & He, J. (2018). Short-term load forecasting based on support vector regression with improved grey wolf optimizer. *Proceedings - 2018 10th International Conference on Advanced Computational Intelligence, ICACI 2018*, 807–812. https://doi.org/10.1109/ICACI.2018.8377565

Jiang, Tianhua-Zhu, H., & Deng, G. (2020). Improved African buffalo optimization algorithm for the green flexible job shop scheduling problem considering energy consumption. *Journal of Intelligent and Fuzzy Systems*, *38*(4), 4573–4589. https://doi.org/10.3233/JIFS-191370

Jungwon, Yu., June, H. Park., & Sungshin, Kim. (2018). A New Input Selection Algorithm Using the Group Method of Data Handling and Bootstrap Method for Support Vector Regression Based Hourly Load Forecasting. *Energies, 11*(11). https://doi.org/10.3390/en11112870

Kari, T., Gao, W., Tuluhong, A., Yaermaimaiti, Y., & Zhang, Z. (2018). Mixed Kernel Function Support Vector Regression with Genetic Algorithm for Forecasting Dissolved Gas Content in Power Transformers. *Energies, 11*(9), 2437. https://doi.org/10.3390/en11092437

Kelo, S., & Dudul, S. (2012). A wavelet Elman neural network for short-term electrical load prediction under the influence of temperature. *International Journal of Electrical Power and Energy Systems*, *43*(1), 1063–1071. https://doi.org/10.1016/j.ijepes.2012.06.009

Kesavan, D., Periyathambi, E., & Chokkalingam, A. (2022). A proportional fair scheduling strategy using multiobjective gradient-based African buffalo optimization algorithm for effective resource allocation and interference minimization. *International Journal of Communication Systems*, *35*(1). https://doi.org/10.1002/dac.5003

Kołodziejczyk, J., & Tarasenko, Y. (2021). Particle Swarm Optimization and Lévy Flight integration. *Procedia Computer Science*, *192*(2019), 4658–4671. https://doi.org/10.1016/j.procs.2021.09.244

Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2017). Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid*. https://doi.org/10.1109/TSG.2017.2753802

Kong, W., Dong, Z. Y., Luo, F., Meng, K., Zhang, W., Wang, F., & Zhao, X. (2018). Effect of automatic hyperparameter tuning for residential load forecasting via deep learning. In *2017 Australasian Universities Power Engineering Conference, AUPEC 2017* (pp. 1–6). https://doi.org/10.1109/AUPEC.2017.8282478

Krisnawati, M., Sibarani, A. A., Mustikasari, A., & Aulia, D. (2020). African Buffalo Optimization for Solving Flow Shop Scheduling Problem to Minimize Makespan. *IOP Conference Series: Materials Science and Engineering*, *982*(1). https://doi.org/10.1088/1757-899X/982/1/012061

Kristjanpoller, W., & Minutolo, M. C. (2018). A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. *Expert Systems with Applications*, *109*, 1–11. https://doi.org/10.1016/j.eswa.2018.05.011

Kumar, S., Hussain, L., Banarjee, S., & Reza, M. (2018). Energy Load Forecasting using Deep Learning Approach-LSTM and GRU in Spark Cluster. *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 1–4. https://doi.org/10.1109/EAIT.2018.8470406

Lachut, D., Banerjee, N., & Rollins, S. (2015). Predictability of energy use in homes. *2014 International Green Computing Conference, IGCC 2014*, 1–10. https://doi.org/10.1109/IGCC.2014.7039146

Lai, X., & Zhou, Y. (2019). An adaptive parallel particle swarm optimization for numerical optimization problems. *Neural Computing and Applications*, *31*(10), 6449–6467. https://doi.org/10.1007/s00521-018-3454-9

Lang, K., Zhang, M., Yuan, Y., & Yue, X. (2018). Short-term load forecasting based on multivariate time series prediction and weighted neural network with random weights and kernels. *Cluster Computing*, 1–9. https://doi.org/10.1007/s10586-017-1685-7

Li, M., Geng, J., Hong, W., & Zhang, Y. (2018). Hybridizing Chaotic and Quantum Mechanisms and Fruit Fly Optimization Algorithm with Least Squares Support Vector Regression Model in Electric Load Forecasting. *Energies*, *11*(9), 2226. https://doi.org/10.3390/en11092226

Li, M. W. M.-W., Geng, J., Wang, S., & Hong, W. C. W.-C. (2017). Hybrid chaotic quantum bat algorithm with SVR in electric load forecasting. *Energies*, *10*(12), 1–18. https://doi.org/10.3390/en10122180

Li, S., Fang, H., & Liu, X. (2018). Parameter optimization of support vector regression based on sine cosine algorithm. *Expert Systems with Applications*, *91*, 63–77. https://doi.org/10.1016/j.eswa.2017.08.038

Li, S. Y., Wang, S. M., Wang, P. F., Su, X. L., Zhang, X. S., & Dong, Z. H. (2018). An improved grey wolf optimizer algorithm for the inversion of geoelectrical data. *Acta Geophysica*, *66*(4), 607–621. https://doi.org/10.1007/s11600-018-0148-8

Li, Y., Che, J., & Yang, Y. (2018). Subsampled Support Vector Regression Ensemble for Short Term Electric Load Forecasting. *Energy*, *164*, 160–170. https://doi.org/10.1016/j.energy.2018.08.169

Lim, M.-P., Habibullah, S., & Lee, H.-A. (2005). Non-linear Dependence in the Malaysian Stock Market. *Pertanika Journal Of*, *13*(1), 23.

Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, *68*(June 2017), 63–80. https://doi.org/10.1016/j.engappai.2017.10.024

Lu, H., Wang, X., Fei, Z., & Qiu, M. (2014). The Effects of Using Chaotic Map on Improving the Performance of Multiobjective Evolutionary Algorithms. *Mathematical Problems in Engineering*, *2014*, 1–16. https://doi.org/10.1155/2014/924652

Ludwig, S. A., & Schoene, T. (2012). Matchmaking in multi-attribute auctions using a genetic algorithm and a particle swarm approach. *Studies in Computational Intelligence*, *383*(November), 81–98. https://doi.org/10.1007/978-3-642-24696-8_5

Luferov, V., Borisov, V., Stefantsov, A., Solopov, R., & Fedulova, S. (2017). Method for electric power load forecasting taking into account meteorological factors based on fuzzy models. *International Journal of Applied Engineering Research*, *12*(23), 13834–13842.https://www.scopus.com/inward/record.uri?eid=2-s2.0-85052669851&partnerID=40&md5=ec715fc6307dc26b833a485bac1c5a1d

M. O., A. (2015). Detection of Non-Linearity in the Time Series Using BDS Test. *Science Journal of Applied Mathematics and Statistics*, *3*(4), 184. https://doi.org/10.11648/j.sjams.20150304.13

Madrid, E. A., & Antonio, N. (2021). Short-term electricity load forecasting with machine learning. *Information (Switzerland)*, *12*(2), 1–21. https://doi.org/10.3390/info12020050

Mahmoudi, N., Orouji, H., & Fallah-Mehdipour, E. (2016). Integration of Shuffled Frog Leaping Algorithm and Support Vector Regression for Prediction of Water Quality

Parameters. *Water Resources Management*, *30*(7), 2195–2211. https://doi.org/10.1007/s11269-016-1280-3

Marinakis, Y., Marinaki, M., & Dounias, G. (2011). Honey bees mating optimization algorithm for the Euclidean traveling salesman problem. *Information Sciences*, *181*(20), 4684–4698. https://doi.org/10.1016/j.ins.2010.06.032

Masum, S., Liu, Y., & Chiverton, J. (2018). Multi-step Time Series Forecasting of Electric Load Using Machine Learning Models. *Artificial Intelligence and Soft Computing. ICAISC 2018. Lecture Notes in Computer Science, Vol 10841. Springer, Cham*, *10841*. https://doi.org/10.1016/j.wpi.2016.08.001

Mat Daut, M. A., Hassan, M. Y., Abdullah, H., Rahman, H. A., Abdullah, M. P., & Hussin, F. (2017). Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review. *Renewable and Sustainable Energy Reviews*, *70*, 1108–1118. https://doi.org/10.1016/j.rser.2016.12.015

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Megala, T., Vivekanandan, K., & Kashyap, M. (2016). Clonal selection algorithm with adaptive lévy mutation operator. *ACM International Conference Proceeding Series*, *25-26-Augu*. https://doi.org/10.1145/2980258.2982125

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mishra, S. (2022). An Optimized Gradient Boost Decision Tree Using Enhanced African Buffalo Optimization Method for Cyber Security Intrusion Detection. *Applied Sciences (Switzerland)*, *12*(24). https://doi.org/10.3390/app122412591

Mocanu, E., Nguyen, P. H., Gibescu, M., & Kling, W. L. (2016). Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks*, *6*, 91–99. https://doi.org/10.1016/j.segan.2016.02.005

Mocanu, E., Nguyen, P. H., Gibescu, M., Larsen, E. M., & Pinson, P. (2016). Demand forecasting at low aggregation levels using Factored Conditional Restricted Boltzmann Machine. *19th Power Systems Computation Conference, PSCC 2016*. https://doi.org/10.1109/PSCC.2016.7540994

Mohanad, S. A., Ravinesh, C. D., & Yan, L. (2018). Particle Swarm Optimized–Support Vector Regression Hybrid Model for Daily Horizon Electricity Demand Forecasting Using Climate Dataset. *E3S Web of Conferences*, *64*(4), 08001. https://doi.org/10.1051/e3sconf/20186408001

Moon, J., Park, J., Hwang, E., & Jun, S. (2018). Forecasting power consumption for higher educational institutions based on machine learning. *Journal of Supercomputing*, *74*(8), 3778–3800. https://doi.org/10.1007/s11227-017-2022-x

Mustaffa, Z. B. (2014). *Enhanced Artificial Bee Colony-Least Squares Support Vector Machines Algorithm for Time Series Prediction*.

Mustaffa, Z., Mohd Herwan, S., Ernawan, F., & Yusof, Y. (2018). Dengue Outbreak Prediction : Hybrid Meta-heuristic Model. *IEEE Computer Society,* 271–274.

Mustaffa, Z., Sulaiman, M. H., & Yusof, Y. (2015). An Application of Grey Wolf Optimizer for Commodity Price Forecasting. *Applied Mechanics and Materials*, *785*, 473–478. https://doi.org/10.4028/www.scientific.net/AMM.785.473

Mustaffa, Z., & Yusof, Y. (2011). A Comparison of Normalization Techniques in Predicting Dengue Outbreak. *International Conference on Business and Economics Research*, *1*, 345–349.

Odili, B. J., Mohmad Kahar, M. N., S, A., & M, A. (2017). Tutorials on African Buffalo Optimization for Solving the Travelling Salesman Problem. *International Journal of Software Engineering and Computer Systems*, *3*(March), 120–128. https://doi.org/10.15282/ijsecs.3.2017.8.0030

Odili, J. B., Kahar, M. N. M., & Anwar, S. (2015). African Buffalo Optimization: A Swarm-Intelligence Technique. *Procedia Computer Science*, *76*, 443–448. https://doi.org/10.1016/j.procs.2015.12.291

Odili, J. B., Kahar, M. N. M., & Noraziah, A. (2017). Parameters-Tuning of PID controller for automatic voltage regulators using the African buffalo optimization. *PLoS ONE*, *12*(4), 1–17. https://doi.org/10.1371/journal.pone.0175901

Odili, J. B., Kahar, M. N. M., Noraziah, A., Zarina, M., & Haq, R. U. (2017a). Performance Analyses of Nature-inspired Algorithms on the Traveling Salesman's Problems for Strategic Management. *Intelligent Automation and Soft Computing*, *8587*(June), 1–11. https://doi.org/10.1080/10798587.2017.1334370

Odili, J. B., Kahar, M. N. M., Noraziah, A., Zarina, M., & Haq, R. U. (2017b). Performance Analyses of Nature-inspired Algorithms on the Traveling Salesman's Problems for Strategic Management. *Intelligent Automation and Soft Computing*, 1–11. https://doi.org/10.1080/10798587.2017.1334370

Odili, J. B., Kahar, M. N., & Noraziah, A. (2016). Solving Traveling Salesman' s Problem Using African Buffalo Optimization, Honey Bee Mating Optimization & Lin-Kerninghan Algorithm. *World Applied Science Journal*, *34*(7), 911–916. https://doi.org/10.5829/idosi.wasj.2016.34.7.329

Odili, J. B., & Mohmad Kahar, M. N. (2016). Solving the Traveling Salesman's Problem Using the African Buffalo Optimization. *Computational Intelligence and Neuroscience*, *2016*. https://doi.org/10.1155/2016/1510256

Odili, J. B., & Noraziah, A. (2018). African buffalo optimization for global optimization. *Current Science*, *114*(3), 627–636. https://doi.org/10.18520/cs/v114/i03/627-636

Odili, J., & Kahar, M. (2015). Numerical Function Optimization Solutions Using the African Buffalo Optimization Algorithm (ABO). *British Journal of Mathematics & Computer Science*, *10*(1), 1–12. https://doi.org/10.9734/bjmcs/2015/17145

Ouahilal, M., Mohajir, M. E., Chahhou, M., & Mohajir, B. E. E. (2017). A Novel Hybrid Model based on Hodrick–Prescott filter and Support Vector Regression Algorithm for Optimizing Stock Market Price Prediction. *Journal of Big Data*, *4*(1). https://doi.org/10.1186/s40537-017-0092-5

Palyulin, V. V., Chechkin, A. V., & Metzler, R. (2014). Levy flights do not always optimize random blind search for sparse targets. *Proceedings of the National Academy of Sciences*, *111*(8), 2931–2936. https://doi.org/10.1073/pnas.1320424111

Pan, J., Zhuang, Y., & Fong, S. (2016). The impact of data normalization on stock market prediction: Using SVM and technical indicators. *Communications in Computer and Information Science*, *652*, 72–88. https://doi.org/10.1007/978-981-10-2777-2_7

Pang, J., He, J., & Dong, H. (2018). Hybrid evolutionary programming using adaptive Lévy mutation and modified Nelder–Mead method. *Soft Computing*. https://doi.org/10.1007/s00500-018-3422-4

Panhalkar, A. R., & Doye, D. D. (2022). Optimization of decision trees using modified African buffalo algorithm. *Journal of King Saud University - Computer and Information Sciences*, *34*(8), 4763–4772. https://doi.org/10.1016/j.jksuci.2021.01.011

Peace Igiri, C., Singh, Y., & Bhargava, D. (2018). An Improved African Buffalo Optimization algorithm using Chaotic map and Chaotic-levy flight. *International Journal of Engineering & Technology*, *7*(4), 4570–4576. www.sciencepubco.com/index.php/IJET

Peng, L. L., Fan, G. F., Huang, M. L., & Hong, W. C. (2016). Hybridizing DEMD and quantum PSO with SVR in electric load forecasting. *Energies*, *9*(3). https://doi.org/10.3390/en9030221

Peng, Y., Xiang, Y., & Zhong, Y. (2013). Quantum-behaved particle swarm optimization algorithm with Lévy mutated global best position. *Proceedings of the 2013 International Conference on Intelligent Control and Information Processing, ICICIP 2013*, 529–534. https://doi.org/10.1109/ICICIP.2013.6568132

Petry, I., Fan, F. M., Siqueira, V. A., Collishonn, W., de Paiva, R. C. D., Quedi, E., de Araújo Gama, C. H., Silveira, R., Freitas, C., & Paranhos, C. S. A. (2023). Seasonal streamflow forecasting in South America's largest rivers. *Journal of Hydrology: Regional Studies*, *49*, 101487. https://doi.org/https://doi.org/10.1016/j.ejrh.2023.101487

Qiang, S., & Brad S., C. (1993). Fuzzy Time Series and its Models. *Fuzzy Sets and Systems*, *54*, 269–277.

Qu, H., & Zhang, Y. (2016). A new kernel of support vector regression for forecasting high-frequency stock returns. In *Mathematical Problems in Engineering* (Vol. 2016). https://doi.org/10.1155/2016/4907654

Ray, P., Panda, S. K., & Mishra, D. P. (2019). Short-term load forecasting using genetic algorithm. *Advances in Intelligent Systems and Computing*, *711*, 863–872. https://doi.org/10.1007/978-981-10-8055-5_76

Rezaee Jordehi, A. (2015). Chaotic bat swarm optimisation (CBSO). *Applied Soft Computing Journal*, *26*, 523–530. https://doi.org/10.1016/j.asoc.2014.10.010

Rezaeian-Zadeh, M., Zand-Parsa, S., Abghari, H., Zolghadr, M., & Singh, V. P. (2012). Hourly air temperature driven using multi-layer perceptron and radial basis function networks in arid and semi-arid regions. *Theoretical and Applied Climatology*, *109*(3–4), 519–528. https://doi.org/10.1007/s00704-012-0595-0

Rumelhart, D. E., Hinton, G. E., & Williams, J. R. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 323–533.

Sadaei, H. J., de Lima e Silva, P. C., Guimarães, F. G., & Lee, M. H. (2019). Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*, 365–377. https://doi.org/10.1016/j.energy.2019.03.081

Sadaei, H. J., Guimaraes, F. G., Jose da Silva, C., Lee, M. H., & Eslami, T. (2017). Short-term load forecasting method based on fuzzy time series, seasonality and long memory process. *International Journal of Approximate Reasoning*, *83*, 196–217. https://doi.org/10.1016/j.ijar.2017.01.006

Salkuti, S. R. (2018). Short-term electrical load forecasting using radial basis function neural networks considering weather factors. *Electrical Engineering*, *100*(3), 1985–1995. https://doi.org/10.1007/s00202-018-0678-8

Sarhani, M., Afia, A. E., & El Afia, A. (2018). Generalization enhancement of support vector regression in electric load forecasting with model selection. *Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications - LOPAL '18*, 1–6. https://doi.org/10.1145/3230905.3230947

Sarhani, M., & El Afia, A. (2015). Electric load forecasting using hybrid machine learning approach incorporating feature selection. *Proceedings of the International Conference on Big Data Cloud and Applications*, *1580*(1992), 1–7. https://doi.org/10.1093/icb/45.5.710

Sayed, W. S., Fahmy, H. A. H., Rezk, A. A., & Radwan, A. G. (2017). Generalized Smooth Transition Map Between Tent and Logistic Maps. *International Journal of Bifurcation and Chaos*, *27*(01), 1730004. https://doi.org/10.1142/s021812741730004x

Sermpinis, G., Stasinakis, C., & Hassanniakalager, A. (2017). Reverse adaptive krill herd locally weighted support vector regression for forecasting and trading exchange traded funds. *European Journal of Operational Research*, *263*(2), 540–558. https://doi.org/10.1016/j.ejor.2017.06.019

Sermpinis, G., Stasinakis, C., Rosillo, R., & de la Fuente, D. (2017). European Exchange Trading Funds Trading with Locally Weighted Support Vector Regression. *European Journal of Operational Research*, *258*(1), 372–384. https://doi.org/10.1016/j.ejor.2016.09.005

Sheeba, R., Sharmila, R., Alkhayyat, A., & Malik, R. Q. (2023). Modified Buffalo Optimization with Big Data Analytics Assisted Intrusion Detection Model. *Computer Systems Science and Engineering*, *46*(2), 1415–1429. https://doi.org/10.32604/csse.2023.034321

Silva, G. C., Silva, J. L. R., Lisboa, A. C., Vieira, D. A. G., & Saldanha, R. R. (2018). Advanced fuzzy time series applied to short term load forecasting. *2017 IEEE Latin American Conference on Computational Intelligence, LA-CCI 2017 - Proceedings*, *2017-Novem*, 1–6. https://doi.org/10.1109/LA-CCI.2017.8285726

Singh, R., & Agarwal, Dr. B. B. (2022). A Hybrid Approach for Detection of Brain Tumor with Levy Flight Cuckoo Search. *Webology*, *19*(1), 5388–5401. https://doi.org/10.14704/web/v19i1/web19361

Singhal, S., Jatana, N., Subahi, A. F., Gupta, C., Khalaf, O. I., & Alotaibi, Y. (2023). Fault Coverage-Based Test Case Prioritization and Selection Using African Buffalo Optimization. *Computers, Materials and Continua*, *74*(3), 6755–6774. https://doi.org/10.32604/cmc.2023.032308

Sinha, A., Tayal, R., Vyas, A., Pandey, P., & Vyas, O. P. (2021). Forecasting Electricity Load With Hybrid Scalable Model Based on Stacked Non-Linear Residual Approach.

Frontiers in Energy Research, *9*(November), 1–17. https://doi.org/10.3389/fenrg.2021.720406

Skare, M., Tomic, D., & Porada-Rochoń, M. (2019). Testing nonlinear dynamics in terms of trade with aggregated data: Implications for economic growth models. *Engineering Economics*, *30*(3), 316–325. https://doi.org/10.5755/j01.ee.30.3.23446

Smola, A. J., & Scholkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, *14*, 199–222. https://doi.org/10.1210/mend.10.7.8813722

Soneji, H., & Sanghvi, R. C. (2014). Towards the improvement of Cuckoo search algorithm. *International Journal of Computer Information Systems and Industrial Management Application*, *6*, 77–88. https://doi.org/10.1109/WICT.2012.6409199

Su, P., Tian, X., Wang, Y., Deng, S., Zhao, J., An, Q., & Wang, Y. (2017). Recent trends in load forecasting technology for the operation optimization of distributed energy system. *Energies*, *10*(9). https://doi.org/10.3390/en10091303

Su, W. H., & Chawalit, J. (2018). Short-term Electricity Load Forecasting in Thailand: an Analysis on Different Input Variables. *IOP Conference Series: Earth and Environmental Science*, *192*. https://doi.org/10.1088/1755-1315/192/1/012040

Subramani, S., Vijayarangan, K. K., & Chenniappan, M. (2023). Improved African Buffalo Optimization-Based Takagi–Sugeno–Kang Fuzzy PI Controller for Speed Control in BLDC Motor. *Electric Power Components and Systems*, *51*(17), 1948–1962. https://doi.org/10.1080/15325008.2023.2210570

Sun, Y., Liu, Y., Zhang, L., Cao, Y., & Zhao, X. (2018). An efficient edge sparse coding approach to ultra-short-term household electricity demand estimation. *IEEJ Transactions on Electrical and Electronic Engineering*, *13*(11), 1586–1594. https://doi.org/10.1002/tee.22723

Suresh, S., & Lal, S. (2016). An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions. *Expert Systems with Applications*, *58*, 184–209. https://doi.org/10.1016/j.eswa.2016.03.032

Sushma, G., Alagarsamy, A., Gopalakrishnan, L., & Vadde, A. R. (2022). Multicriteria Deming Regressive African Buffalo Optimized Mapping for 3D NoC Architecture Design. *Mathematical Problems in Engineering*, *2022*. https://doi.org/10.1155/2022/5084178

Tarkhaneh, O., & Shen, H. (2019). Training of feedforward neural networks for data classification using hybrid particle swarm optimization, Mantegna Lévy flight and neighborhood search. *Heliyon*, *5*(4), e01275. https://doi.org/10.1016/j.heliyon.2019.e01275

Tchuidjan, R., Hamandjoda, O., Mbobda, G., Tabe, M., & Touke, Y. (2014). Improved electric power demand forecasting by adapting the weighted average to the MISMO strategy. *International Journal on Electrical Engineering and Informatics, 6*(2), 237–249.

Tharwat, A., & Hassanien, A. E. (2018). Chaotic antlion algorithm for parameter optimization of support vector machine. *Applied Intelligence*, *48*(3), 670–686. https://doi.org/10.1007/s10489-017-0994-0

Tran, T. H., & Hoang, N. D. (2017). Estimation of algal colonization growth on mortar surface using a hybridization of machine learning and metaheuristic optimization. *Sadhana - Academy Proceedings in Engineering Sciences*, *42*(6), 929–939. https://doi.org/10.1007/s12046-017-0652-6

Tutun, S. (2016). *The Monthly Electricity Dataset for Turkey*. Mendeley. https://doi.org/10.17632/2Y4ZFTPMX4.1

Tutun, S., Chou, C. A., & Caniyilmaz, E. (2015). A new forecasting framework for volatile behavior in net electricity consumption: A case study in Turkey. *Energy*, *93*, 2406–2422. https://doi.org/10.1016/j.energy.2015.10.064

Ugurlu, U., Oksuz, I., & Tas, O. (2018). Electricity Price Forecasting Using Recurrent Neural Networks. *Energies*, *2145*(5), 75–78. https://doi.org/10.3390/en11051255

Ullah Khan, R., Bhattacharyya, B., & Singh, G. (2021). A novel African buffalo optimization for the minimization of cogging torque in modified permanent magnet DC motor. *Sustainable Energy Technologies and Assessments*, *46*. https://doi.org/10.1016/j.seta.2021.101240

Vaza, R. N., Prajapati, R., Rathod, D., & Vaghela, D. (2022). Developing a novel methodology for virtual machine introspection to classify unknown malware functions. *Peer-to-Peer Networking and Applications*, *15*(1), 793–810. https://doi.org/10.1007/s12083-021-01281-5

Velasco, L. C. P., Polestico, D. L. L., Abella, D. M. M., Alegata, G. T., & Luna, G. C. (2018). Day-ahead load forecasting using support vector regression machines. *International Journal of Advanced Computer Science and Applications*, *9*(3), 22–27. https://doi.org/10.14569/IJACSA.2018.090305

Verma, M., Thirumalaiselvi, A., & Rajasankar, J. (2017). Kernel-based models for prediction of cement compressive strength. *Neural Computing and Applications*, *28*, 1083–1100. https://doi.org/10.1007/s00521-016-2419-0

Wahid, F., & Kim, D. H. (2016). A prediction approach for demand analysis of energy consumption using K-nearest neighbor in residential buildings. *International Journal of Smart Home*, *10*(2), 97–108. https://doi.org/10.14257/ijsh.2016.10.2.10

Wang, T., & Wang, W. (2019). Research on a Thangka image classification method based on Support vector machine. *International Journal of Pattern Recognition and Artificial Intelligence*, *33*(12), 1–15. https://doi.org/10.1142/s0218001419540302

Wei, N., Li, C., Duan, J., Liu, J., & Zeng, F. (2019). Daily Natural Gas Load Forecasting Based on a Hybrid Deep Learning Model. *Energies*, *12*(2), 218. https://doi.org/10.3390/en12020218

Wei, N., Li, C., Li, C., Xie, H., Du, Z., Zhang, Q., & Zeng, F. (2018). Short-Term Forecasting of Natural Gas Consumption Using Factor Selection Algorithm and Optimized Support Vector Regression. *Journal of Energy Resources Technology*, *141*(3), 032701. https://doi.org/10.1115/1.4041413

Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, *30*(4), 1030–1081. https://doi.org/10.1016/j.ijforecast.2014.08.008

Xianlong, L., Cheng, X., Shuang, Y., & Tang, Y. M. (2018). Short-term Power Load Forecasting Based on Balanced KNN. *IOP Conference Series: Materials Science and Engineering*, *322*(7). https://doi.org/10.1088/1757-899X/322/7/072058

Xie, H., Ding, M., Chen, L., An, J., Chen, Z., & Wu, M. (2017). Short-term wind power prediction by using empirical mode decomposition based GA-SVR. *Chinese Control Conference*, *(CCC)*, 9175–9180. https://doi.org/10.23919/ChiCC.2017.8028818

Yan, A., Huang, X., & Shao, H. (2017). On the Sparseness and Generalization Capability of Least Squares Support Vector Machines. *Journal of Systems Science and Information*, *3*(3), 279–288. https://doi.org/10.1515/jssi-2015-0279

Yan, X., Zhang, C., Luo, W., Li, W., Chen, W., & Liu, H. (2012). Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm. *International Journal of Computer Science*, *9*(6), 264–271. https://doi.org/10.3205/zma000782

Yang, A., Li, W., & Yang, X. (2019). Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines. *Knowledge-Based Systems*, *163*, 159–173. https://doi.org/10.1016/j.knosys.2018.08.027

Yao, X., & Mao, S. (2023). Electric supply and demand forecasting using seasonal grey model based on PSO-SVR. *Grey Systems*, *13*(1), 141–171. https://doi.org/10.1108/GS-10-2021-0159

Yusof, Y., & Mustaffa, Z. (2015). Time Series Forecasting of Energy Commodity. *International Journal of Engineering Research and Applications*, *I*.

Yusof, Y., & Mustaffa, Z. (2016). A Review on Optimization of Least Squares Support Vector Machine for Time Series Forecasting. *International Journal of Artificial Intelligence & Applications*, *7*(2), 35–49. https://doi.org/10.5121/ijaia.2016.7203

Zaimoğlu, E. A., Yurtay, N., Demirci, H., & Yurtay, Y. (2023). A binary chaotic horse herd optimization algorithm for feature selection. *Engineering Science and Technology, an International Journal*, *44*. https://doi.org/10.1016/j.jestch.2023.101453

Zhang, S., Kuang, F., & Hu, R. (2019). Support Vector Regression with Multi-Strategy Artificial Bee Colony Algorithm for Annual Electric Load Forecasting. *Advances in Intelligent Systems and Computing*, *891*, 576–585. https://doi.org/10.1007/978-3-319-68527-4

Zhang, X. (2018). Short-term load forecasting for electric bus charging stations based on fuzzy clustering and least squares support vector machine optimized by Wolf pack algorithm. *Energies*, *11*(6). https://doi.org/10.3390/en11061449

Zhang, X.-M., Wang, X., Tu, Q., & Kang, Q. (2018). Particle Swarm Optimization Algorithm Based on Combining Global-Best Operator and Levy Flight. *Dianzi Keji Daxue Xuebao/Journal of the University of Electronic Science and Technology of China*, *47*(3), 421–429. https://doi.org/10.3969/j.issn.1001-0548.2018.03.016

Zheng, M., Tang, W., & Zhao, X. (2019). Hyperparameter optimization of neural network-driven spatial models accelerated using cyber-enabled high-performance computing. *International Journal of Geographical Information Science*, *33*(2), 314–345. https://doi.org/10.1080/13658816.2018.1530355

Zhu, H., Jiang, T., & Wang, Y. (2020). Discrete African Buffalo Optimization Algorithm for the Low-carbon Flexible Job Shop Scheduling Problem. *Journal of Advanced Manufacturing Systems*, *19*(4), 837–854. https://doi.org/10.1142/S0219686720500390

Ziel, F. (2015). Modelling and forecasting electricity load using lasso methods. *Proceedings - International Conference on Modern Electric Power Systems, MEPS 2015*. https://doi.org/10.1109/MEPS.2015.7477217

Appendix A

BDS Test Results on Datasets

BDS Test for TOTAL
Date: 08/20/23 Time: 16:22
Sample: 12/16/2006 12/31/2009
Included observations: 1112

Dimension	BDS Statistic	Std. Error	z-Statistic	Prob.
2	0.054404	0.002296	23.69301	0.0000
3	0.089311	0.003642	24.52166	0.0000
4	0.117813	0.004328	27.22046	0.0000
5	0.134427	0.004501	29.86271	0.0000
6	0.138238	0.004332	31.91293	0.0000

Raw epsilon	14736.99		
Pairs within epsilon	870046.0	V-Statistic	0.703611
Triples within epsilon	7.33E+08	V-Statistic	0.533337

Dimension	C(m,n)	c(m,n)	C(1,n-(m-1))	c(1,n-(m-1))	c(1,n-(m-1))^k
2	338713.0	0.549319	433783.0	0.703502	0.494915
3	269638.0	0.438083	433250.0	0.703905	0.348772
4	223200.0	0.363290	432458.0	0.703886	0.245477
5	188419.0	0.307233	431687.0	0.703901	0.172806
6	159184.0	0.260032	431000.0	0.704052	0.121794

Figure A1. Household dataset BDS test result

BDS Test for NET_ELECTRICITY_CONSUMPTION_KWH
Date: 08/20/23 Time: 19:13
Sample: 1 420
Included observations: 420

Dimension	BDS Statistic	Std. Error	z-Statistic	Prob.
2	0.182293	0.002840	64.18835	0.0000
3	0.310368	0.004511	68.79581	0.0000
4	0.398073	0.005368	74.16103	0.0000
5	0.462342	0.005589	82.72568	0.0000
6	0.507087	0.005383	94.19301	0.0000

Raw epsilon	6.12E+09		
Pairs within epsilon	124444.0	V-Statistic	0.705465
Triples within epsilon	39025632	V-Statistic	0.526747

Dimension	C(m,n)	c(m,n)	C(1,n-(m-1))	c(1,n-(m-1))	c(1,n-(m-1))^k
2	59749.00	0.682292	61922.00	0.707106	0.499999
3	58080.00	0.666414	61771.00	0.708765	0.356047
4	56655.00	0.653189	61643.00	0.710697	0.255116
5	55766.00	0.646038	61508.00	0.712558	0.183696
6	55058.00	0.640917	61439.00	0.715197	0.133830

Figure A2. Turkey electricity consumption dataset BDS result

BDS Test for TOTAL
Date: 08/20/23 Time: 19:19
Sample: 1/11/2016 17:00 5/27/2016 18:00
Included observations: 19735

Dimension	BDS Statistic	Std. Error	z-Statistic	Prob.
2	0.139832	0.000987	141.6880	0.0000
3	0.232171	0.001525	152.2148	0.0000
4	0.288072	0.001768	162.9154	0.0000
5	0.319218	0.001795	177.8236	0.0000
6	0.334430	0.001687	198.2529	0.0000

Raw epsilon	68.02395		
Pairs within epsilon	2.65E+08	V-Statistic	0.681061
Triples within epsilon	4.10E+12	V-Statistic	0.533163

Dimension	C(m,n)	c(m,n)	C(1,n-(m-1))	c(1,n-(m-1))	c(1,n-(m-1))^k
2	1.18E+08	0.603746	1.33E+08	0.681112	0.463914
3	1.07E+08	0.548241	1.33E+08	0.681179	0.316070
4	98005127	0.503453	1.33E+08	0.681243	0.215381
5	90690703	0.465926	1.33E+08	0.681225	0.146707
6	84537550	0.434358	1.33E+08	0.681209	0.099927

Figure A3. Appliances electricity consumption dataset BDS test result

BDS Test for NAT_DEMAND
Date: 08/20/23 Time: 21:42
Sample: 1/03/2015 01:00 6/27/2020 00:00
Included observations: 48048

Dimension	BDS Statistic	Std. Error	z-Statistic	Prob.
2	0.154029	0.000212	726.0282	0.0000
3	0.248764	0.000335	743.4459	0.0000
4	0.302156	0.000395	764.5519	0.0000
5	0.328953	0.000408	805.3710	0.0000
6	0.339983	0.000391	870.5993	0.0000

Raw epsilon	295.1069		
Pairs within epsilon	1.62E+09	V-Statistic	0.702037
Triples within epsilon	5.72E+13	V-Statistic	0.516107

Dimension	C(m,n)	c(m,n)	C(1,n-(m-1))	c(1,n-(m-1))	c(1,n-(m-1))^k
2	7.47E+08	0.646875	8.10E+08	0.702030	0.492845
3	6.86E+08	0.594753	8.10E+08	0.702027	0.345988
4	6.29E+08	0.545043	8.10E+08	0.702023	0.242887
5	5.76E+08	0.499459	8.10E+08	0.702017	0.170505
6	5.30E+08	0.459674	8.10E+08	0.702011	0.119691

Figure A4. Panama electricity consumption dataset BDS test result