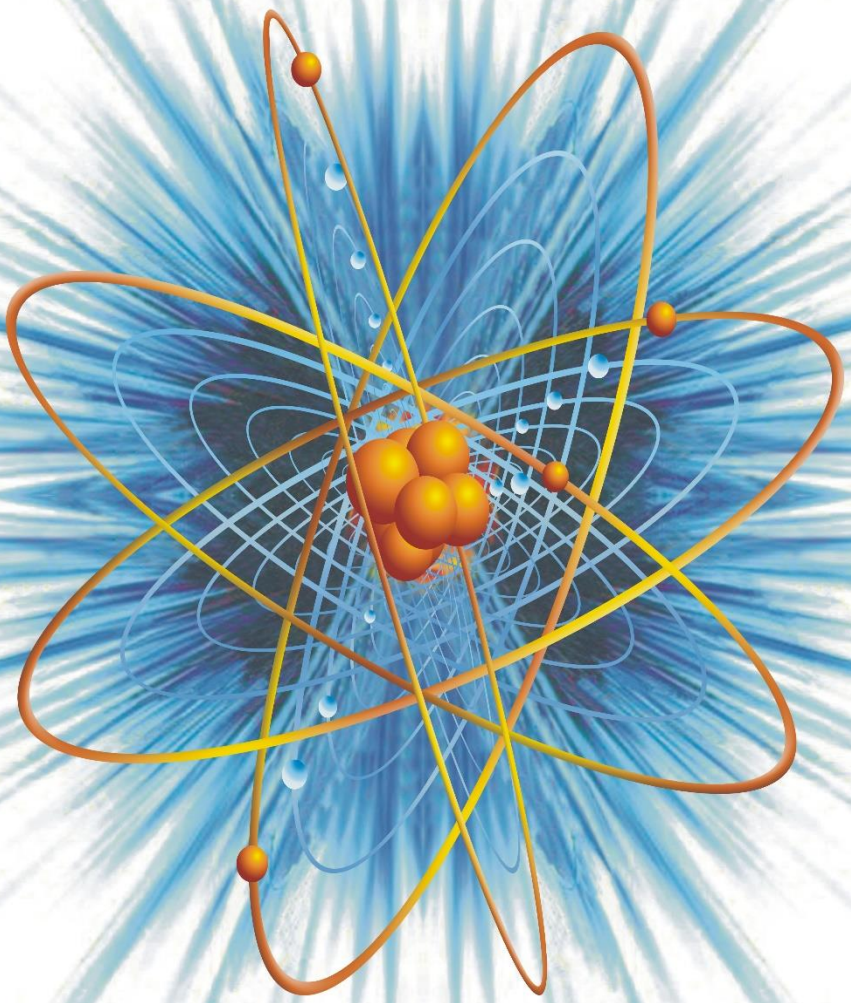
The Nucleus

An Open Access International Scientific Journal



Vol. 61, No. 1, 2024

ISSN 0029-5698 (Print)

eISSN 2306-6539 (Online)

The Nucleus

An international journal devoted to all branches of natural and applied sciences

Website: *www.thenucleuspak.org.pk* **E-mail:** *editorialoffice@thenucleuspak.org.pk*
Phone: +*92-51-9248429* **Fax: +***92-51-9248808*

Editorial Office, The Nucleus

PINSTECH, Nilore, 45650, Islamabad, Pakistan

Printed at

PINSTECH Printing Press

The Nucleus

Aims and Scope: The Nucleus is an open access multidisciplinary peer-reviewed academic journal. It offers a platform for the scientists and engineers to publish their recent research of high scientific values in all areas of natural, applied and management sciences at international level. The journal is being published electronically as well as in paper version. It is easily accessible, free of charge and is being distributed widely.

Abstracting and Indexing: The journal is being abstracted and indexed by Chemical Abstracts, Citefactor, Biological Abstracts, INIS Atom Index, Bibliography of Agriculture (USA), The Institution of Electrical Engineers Publication, Virology Abstracts (England) and Pakistan Science Abstracts. The journal is recognized by the Higher Education Commission of Pakistan (Category Y).

ISSN and eISSN: The international standard serial numbers (ISSN) for The Nucleus are [0029-5698 (Print) and 2306-6539 (Online)].

The Nucleus is published at Pakistan Institute of Nuclear Science & Technology, Islamabad, on behalf of the Pakistan Atomic Energy Commission.

Published since 1964

The Nucleus
ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

CONTENTS

The Nucleus ***61****, No. 1 (2024) 1-9*

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Emotion Detection for Cryptocurrency Tweets Using Machine Learning Algorithms

Bushra Fareed[1], Mujeeb Ur Rehman[2*], Mumtaz Ali Shah[2], Akbar Hussain[2], Khudija Bibi[3]

[1]*Khwaja Fareed University of Engineering & Information Technology, Rahim Yar Khan, Pakistan*

[2]*University of Management and Technology, Sialkot, Pakistan*

[3]*International Islamic University, Islamabad, Pakistan*

ABSTRACT

Cryptocurrencies, functioning as digital currencies, undergo regular fluctuations in the present market, reflecting the emotional aspect of the cryptocurrency realm. It is a well-established fact that sentiment is linked to Bitcoin and Ethereum values, employing a Twitter-based strategy to predict changes. While prospective Bitcoin returns do not display a correlation with emotional variables, indicators of emotions tend to anticipate Bitcoin exchange volume and return volatility. Emotions wield an influence over a broad spectrum of financial investor returns, thereby, potentially affecting market dynamics by triggering significant price shifts. The research delves into gauging emotional factors extracted from 2,050,202 posts on Bitcointalk.org, investigating how these emotions impact Bitcoin's price fluctuations. We have used a unified dataset named 'data F' in which all categories of emotions are consolidated. Subsequently, data preprocessing steps are implemented to cleanse the dataset. Two feature engineering techniques, namely TF-IDF and BoW are employed. The research explores ten supervised machine learning (ML) models as classifiers, with four of these models (LR, Stochastic Gradient Descent, SVM and GB) yielding the highest accuracy at 0.93%.

Keywords: *Emotion, Twitter, Bitcoin, Ethereum, Cryptocurrencies*

1. Introduction

Over the course of an extensive period, artificial intelligence (AI) has evolved from a concept relegated to the realm of fiction to a practical force in the real world, thanks to the availability of adequate computing power for execution. Cryptography plays a pivotal role in safeguarding cryptocurrencies and their transactions [1]. Unlike traditional currencies reliant on central financial institutions, cryptocurrencies operate on the principle of decentralized control. Consequently, cryptocurrencies enable electronic money transfers without the need for intermediaries or conventional financial establishments. With their inherent characteristics of being uncontrollable and untraceable, the cryptocurrency industry has witnessed rapid growth within a short span of time. Virtual currencies are progressively being integrated into economic transactions across various domains. This field has garnered significant attention and a multitude of scholars are scrutinizing cryptocurrencies for diverse purposes, including financial prognostications and more. Researchers have displayed escalating interest in the commercial applications of cryptocurrencies. However, the utility of cryptocurrency and its associated technologies extends beyond the financial spheres. Numerous information technology (IT) courses encompass cryptocurrency technologies that hold the potential to devise novel and efficient methodologies for managing bitcoin, other cryptocurrencies, their price volatility, and related technologies. This article provides an overarching view of cryptocurrency price prediction research spanning the years from 2010 to 2020 [16]. It encompasses seminal studies concerning the prediction of cryptocurrency prices, incorporating both ML methods and statistical approaches. Furthermore, this study delves into datasets, trends, research methodologies and forecasting techniques, subsequently exploring uncharted territories within the realm of bitcoin price prediction research.

From May 2018 onwards, the two most prominent cryptocurrencies, as measured by market capitalization, held a combined total value of approximately $160.9 billion [2]. Out of this, bitcoin alone accounted for $115 billion of the total value. These digital currencies are believed to have value, functioning as real currencies and serving as investment opportunities for certain investors. Both of these currencies experienced significant fluctuations in value within a short time span. In the midst of 2017, the value of a single bitcoin surged by an astonishing two thousand percent, soaring from $863 on January 9 to $17,550 on December 11 of the same year. Merely eight weeks later, by February 5, 2018, the price of a single bitcoin had exceeded $79,643, again marking a two thousand percent increase [12]. The input for the ML models, specifically decision tree (DT), logistic regression (LR), and random forest (RF), involves utilizing user reviews about cryptocurrencies. The objective is to predict the emotional tones of these reviews, namely: joy, sadness, hatred, fear, anger or greed. The dataset employed in this research was sourced from Kaggle and it underwent preprocessing using the natural language toolkit (NLTK) in Python. NLTK was chosen due to its effectiveness in handling human language data for statistical natural language processing tasks. It facilitated the removal of stop words and other text cleaning procedures. Additionally, the text was converted to lowercase and underwent further preprocessing steps including tokenization, stop word removal, porter stemming and more. Upon completing the preprocessing phase, text feature extraction techniques such as term frequency and inverse document frequency were applied. The bag of words (BoW) approach was utilized to identify the most relevant features. Subsequently, the dataset was divided

*Corresponding author: mujeeb.rehman.pak@gmail.com

into two segments: a training set (80%) and a testing set (20%). To conduct the classification of reviews into emotional categories (joy, sadness, fear, hatred, anger and greed), a range of classification algorithms were employed. These included random forest (RF), decision tree (DT), stochastic gradient descent classifier, K-nearest neighbors (KNN), Ada boost classifier (AC), Gaussian Naive Bayes (GNB) and extra tree classifier (ETC). Notably, the first seven models are based on tree ensemble methods, while logistic regression (LR) was used as a distinct classification approach.

The fundamental aspects of employing this approach encompass:

- Categorization of data into six emotional classes. This aids individuals involved in cryptocurrency in comprehending people's emotions, thus, aiding their decision-making. Diverse ML models were chosen for this study, each with distinct parameters. These parameters were fine-tuned through empirical methods to attain optimal levels of accuracy, precision, recall and F1-score.
- Data can be segmented into six emotional types: joy, sadness, fear, hate, as well as greed and anger. This categorization is based on user reviews.
- Application of preprocessing techniques, including converting to lowercase, replacing characters from A to Z, splitting, eliminating stop words, applying porter stemmer, cleansing tweet reviews and utilizing effective learning models.
- Implementation of techniques like: term frequency and inverse document frequency (TF-IDF) and BoW through feature engineering.

2. Related Work

This section provides a concise evaluation of cryptocurrency price prediction methods. The related research can be categorized into three main types: (1) assessment of social media influence on financial markets, including cryptocurrency markets; (2) employment of ML for predicting cryptocurrency prices; and (3) utilization of extensive data frameworks for financial market prediction.

Daniel et al. investigate decision-making growth and risk through alternate models in prospect theory [2, 3]. Typically seen in financial psychology, these models depict a range of sentiments that substantially impact a financial agent's decision-making process, consequently leading to a consistent pattern of price projection [4]. These insights thus pave the way for employing methods like sentiment analysis to identify patterns affecting asset prices.

Regarding media's evolution, particularly on social platforms and their impact on user sentiment in financial sectors, Tetlock's study [5] distinguishes high negativity on social platforms as indicative of downward market pressure, with abnormally high or low negativity predicting high trading volume. Moreover, the majority of consumers resort to social networks for purchase decisions, yielding investigations that explore connections between sentiment in media (such as reviews) and various financial sectors. Heaton's research [6] retrieves, extracts and examines media's effects on financial markets. By constructing a sentiment analysis lexicon specific to the financial domain, the study achieved 70.59 percent accuracy in short-term stock market trend prediction. Stock performance prediction also extends to comment board analysis [7] uncovering the 'topic sentiment' feature reflecting emotions tied to industry-specific subjects and stock projections. This approach has achieved a 2.07 percentage point performance gain over one year with 18 trades, solely using historical prices. Similarly, Alan et al. linked tweet sentiment scores to stock prices [8], achieving an 86.7 percent accuracy in Dow Jones industrial average prediction using a self-constructed fuzzy classifier network. The ensuing increase in cryptocurrency speculation led to efforts to forecast price fluctuations [9]. Jafar et al. achieved successful predictions of price variations for Ethereum, Bitcoin, and Litecoin [10] by leveraging news and social media data annotated with actual prices. The method precisely anticipated price changes, achieving 43.9 percentage points accuracy for price increases and 61.9 percentage points accuracy for price decreases in bitcoin forecasts. Additionally, Twitter sentiment and Google trends data were used for bitcoin and ethereum price prediction [2]. The approach incorporated tweet volume and established a correlation with cryptocurrency prices. Nomiizz further expanded on this, utilizing sentiment analysis of tweets to establish a connection with bitcoin prices [11, 12]. Tweets underwent preprocessing to remove non-alphanumeric characters and were analyzed using the valence aware dictionary and sentiment reasoner (VADER) to categorize them as positive, neutral, or negative. Compound sentiment scores were then correlated with bitcoin prices across different time intervals. Nomiizz's approach extends our previous discussions, presenting a refined forecasting model for bitcoin prices over specified intervals.

Additionally, ML is a pivotal tool for cryptocurrency price forecasting. In a notable instance, the authors of [13] contrast bitcoin price prediction methods by benchmarking traditional auto-regressive integrated moving average (ARIMA) and ML-based neural network auto-regressive (NNAR) models against historical price data. Similarly, Lucey et al. [14] introduce classifiers employing a hierarchy of deep ML models for cryptocurrency price forecasting. This hierarchy includes a multi-layer perceptron (MLP), a basic recurrent neural network (RNN) and a long short-term memory (LSTM) network, specifically designed for complex sequence prediction. Notably, this study expands prediction methodologies by incorporating the impact of social media and applying online learning for better management. This approach accommodates an indefinite timeline for data generation. Consequently, our treatment of the problem aligns with a big data perspective, utilizing substantial data tools to ensure scalability and performance. We draw inspiration from Munim et al. [15] who utilize Apache Spark to analyze market trends on the foundation of social network data and historical

prices. Naila Aslam and her team engage in cryptocurrency tweet detection and sentiment analysis, employing five emotion classes - happy, sad, fear, angry and surprise - utilizing techniques like BoW, TFIDF and Word2Vec [16]. Table 1 presents an extensive summary of various models utilized on a range of sentiment reviews gathered from diverse sources.

Table 1. Summary of Related Work

Ref. No.	Year	Models	Focus	Restriction	Dataset Size	Sources
2	2013	Descriptive model	Prospect Theory	No ML models.	N/A	N/A
3	2017	Efficient Market Hypothesis (EMH)	Feelings affect Investor Decision-Making	No ML models.	N/A	psychologists and behavioral economists
4	2007	Linear combination	Giving Content to Investor Sentiment	This article only discusses interactions between media content and stock market activity.	77 × 77	Dow Jones Newswires, Wall Street Journal column daily (1889 to 1897)
5	2017	Kaiser-Meyer-Olkin (KMO) and Bartlett's Test	Role of Google and Social Networks on Consumers' Buying Behavior	This article only explains 61.53% variety.	160	Google and Social Networks
6	2018	Hybrid Model (ARIMA & LSTM)	Predicting the Effects of News Sentiments on the Stock Market	This article only using news sentiments achieved only 70.59% accuracy in the short-term movement of stock price.	6 Months	specific news articles
7	2017	Support Vector Machine (SVM)	Sentiment Analysis on social media for Stock Movement Prediction	In this article, only a few specified numbers of topics and sentiments beforehand for Joint Sentiment Topic (JST)-based and the Latent Dirichlet Allocation (LDA).	1 Year	social media
8	2014	Dow Jones Industrial Average (DJIA)	Trading on Twitter	No ML models.	3,475,428	Tweets
9	2005	DJIA & Self-Organizing Fuzzy Neural Network	Role of Twitter Mood on the Stock Market	This study did not use ML models.	June 2009 to December 2009	Tweets
10	2017	LR, Bernoulli Naive Bayes & NB	Cryptocurrency Price Prediction Using News and Social Media Sentiment	This article finds the percentage of 67 days increases and decreases the price of bitcoin & Ethereum.	67 Days	Crypto Market
11	2018	Sentiment analysis	Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis	No ML models.	30,420,063	tweets and Google Trends data
12	2007	Sentiment analysis	Role of Twitter Sentiment Analysis in Bitcoin Price Fluctuation	No ML models.	N/A	Tweets
13	2021	VADER, vaderSentiment, Polarity classification & lexicon-based approach	Drabble/TwitterSentimentAndCryptocurrencies	Miss-tuning of hyperparameters	828'338	Tweets
14	2019	ARIMA & NNAR	Next-Day Bitcoin Price Forecast	This article only uses two test sample forecast periods.	Two Sample	Crypto Market
15	2018	MLP& RNN, LSTM	Deep Neural Network for Cryptocurrencies Price Prediction	In this article, use learns long dependencies.	1 Year	Crypto Market
16	2018	Hybrid model-based (Apache Spark & Hadoop HDFS)	Stock Market Real-Time Recommender Model Using Apache Spark Framework	In this article historical price is not used as the primary factor for a prediction of the stock market trend.	1-2-2013 to 30-6-2016	interval
17	2022	LSTM and GRU	Sentiment Analysis and Emotion Detection on Cryptocurrency	y Work five Classes	40,000	Tweets

Table 2 Sample Data

A	B	C
0	Crypto Panic: US Securities Watchdog Says Desperate Kik Turned to Blockchain Tokens https://t.co/CR3tmIuuZ8 Crypto https://t.co/hmEsmDINxh	Fear
1	Earning #cryptocurrency for selling my stuff on @Listia! Join me using code "DGWCDX" for an extra 100 XNK. I just l https://t.co/cYSFzxWZx5	Fear
3963	Let us stop annoying the big exchanges and screw ourselves over.. this is not helping anyone	Hate
8739	Happy Children's Day. Visit https://t.co/2OKeAHtIPH to buy a cryptocurrency and pay with Naira. https://t.co/hUItHTdnkt	Joy
12488	RT @CrptoKryptonite: @blockonix_com @Tronfoundation @Cardano @block_one_ @creditscom will be period. DYOR... low trans fees, most scalable	Sad
15051	Coinbase's crypto debit card arrives in 6 more countries #Cryptocurrencies #bitcoin #crypto #cryptocurrency https://t.co/x3E1Zmj8dt	Greed
21572	RT @jonhumbert: 1) Seditious traitor 2) Seditious traitor with a nickname 3) Total unit/hero 4) Angry boi/hero 5) Seditious traitor who was traitor	Anger

3. Methods and Material

In this study, different ML methods were used for this analysis. Various experiments were investigated using different methods and techniques, classification flowcharts, diagrams, evolutionary parameters and more. Several classifiers were evaluated for this purpose; however, the best classification was the voting classification.

The dataset was first downloaded from Kaggle (https://www.kaggle.com/datasets/huseinzol05/twitter-emotion-cryptocurrency). It was processed by removing irrelevant elements and divided into test and train datasets. The size of the test data set weighed 20% and the size of the training dataset is 80% weighed. The next step is the feature engineering kept in the training package. The test data set was forecast using classification. After the set of data was predicted, evolution parameters were applied to it to achieve faultless accuracy. The following parameters have been used to model evolution: precision, recall, F1-score, and accuracy.

The data for this experiment is taken from Kaggle, which has a large number of opposing tweets. The type of the data file is JavaScript Object Notation (JSON type of file is an open standard format use to share the data, it used human-readable text to save and share the data). The dataset contains six different types of files such as: joy.json, sadness.json, fear.json, hateful.json, anger.json and greed.json. The size of dataset files like joy.json has (54565) reviews, the size of sadness.json has (9765) reviews, the size of fear.json is (50053) reviews, the size of hateful.json is (10206) reviews, the size of anger.json is (3596) reviews and the last file size of greed.json is (200177) reviews. We append all these six files in one single file name dataF and the total size of the dataF file is (328362) reviews. We take target all these files like joy.json takes targets as Joy and sadness.json is sad and fear.json takes target fear, in hateful.json takes targets hate and anger.json takes targets anger and also takes from greed.json is greed reviews. The dataset used in this study is shown in the following Table 2.

Column A of Table 2 shows the number of rows and column B shows the tweets about the cryptocurrency and column C show the target values.

Table 3 shows the total number of data in each file.

Table 3. Target Reviews of Cryptocurrencies

Joy	Sad	Fear	Hate	Greed	Anger
54565	9765	50053	10206	200177	3596

3.1 Engineering of Feature

Engineering of features used to find efficient features from the dataset to use training the models or another way, the best feature selected from the original dataset [17] . Robertson [18] finishes that engineering features can raise the working of ML models. ‘‘refuse in refuse out’ is a mostly saying in ML. Another way, more instructive data can create desirable results. Engineering of feature help to increase the accuracy of the raw dataset.

3.1.1 Term Frequency and Inverse Document Frequency (TF-IDF)

Term frequency-inverse document frequency is used to find and select features in the dataset. Text analysis and music information retrieval use to term frequency-inverse document frequency [19]. The weight is assigned to every word in the document using TF-IDF [15, 20]. Any term or word that has a higher weight means more importance than other terms [21]. This formula is used to find every word or term weight as mentioned in Equation 1.

$$W_{i,j} = TF_{i,j}(\frac{N}{D_{f,t}}) \quad (1)$$

In Equation 1 where N shows the number of documents and the $D_{f,t}$ shows the total number of words in the document.

3.1.2 Bag of Words (BoW)

A bag of words is a simple technique in which we convert simple text data into numeric form. On this numeric form data, we assign the frequency to each word in the data. Frequency depends on how many times the particular word

comes in the corpus on the basis of word frequency we determine the importance of each word in sentimental analysis. This formula is used to find every word or term weight as mentioned in the equation.

3.2 Supervised Machine Learning Methods

In supervised ML, this study discusses the ML models and also shows the algorithms and their hyper parameters. ML uses to apply and implement algorithms by the Scikit-learn library and NLTK [22]. ML-supervised algorithms are mostly used for classification and regression [23]. We use RF, LR, DT, KNN, Ada Boost Classifier, STOCHASTIC GRADIENT DESCENT Classifier, ET, GNB and Gradient Boosting Classifier.

Table 4. ML Models and Hyper-parameters

ML models	Hyperparameters	Using
LR	Multi-class="multinomial"	In LR only work binary class that way we use Multi-class="multinomial" work for multiple classes
RF	n-estimators=50	Rang of n-estimators is 50 to 500 but we use only 50, which gives us the best results
DT	Random-state=0	For solving the problems for train and test split that way we use Random-state=0
KNN	n-neighbors=5	For the range of n-neighbors 1 to 5, we use the 5 that way we get the best results
SVM	kernel='linear'	is used when the data is linearly separable, that is, it can be separated by a single line
Ada Boost Classifier	n-estimators=50, learning-rate=1	In the range of 0.0 to 1.0 learning rate we use 1 for the results
GNB	Priors=none, var_smoothing= 1 e-09	The portion of the largest variance of all elements that is added to the variances for calculation stability
ExtraTree Classifier	Random-state=0	For solving the problems for train and test split that way we use Random-state=0
Stochastic Gradient Descent Classifier	Random-state=0	For solving the problems for train and test split that way we use Random-state=0
Gradient Boosting Classifier	Random-state=0	For solving the problems for train and test split that way we use Random-state=0

3.2.1 Random Forest (RF)

Random forests or random decision forests are an ensemble getting-to-know approach for classification, regression and other obligations that perform by using building a large number of selections at schooling time and outputting the class this is the mode of the training (classification) or suggesting/averaging prediction (regression) of the person branches.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(fi - yi)^2 \quad (2)$$

In Equation 2, where N is the wide variety of information factors, fi is the value returned with the aid of the version and yi is the actual cost for statistics point i.

3.2.2 Logistics Regression (LR)

Logistic regression is a statistical analysis technique used to expect an information price based totally on prior observations of an information set. A logistic regression version predicts a dependent statistics variable by using studying the connection among one or greater current independent variables.

3.2.3 Gaussian NB (GNB)

A Gaussian Naive Bayes set of rules is a unique kind of NB set of rules. It is in particular used while the functions have continuous values. It is also presumed that each one of the functions are following a Gaussian distribution, particularly the everyday distribution. A Gaussian classifier is a generative technique in the sense that it tries to version class posterior as well as enter class-conditional distribution. Consequently, we can generate new samples in the entering area with a Gaussian classifier.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (3)$$

In Equation 3, A and B are two events where P(A|B) or P(B|A) is the probability of the events A and B. The independent probability P(A) of A and P(B) of B.

3.2.4 Extra Tree Classifier (ETC)

Extra Tree Classifier (ETC) is a gathering learning strategy on a very basic level based on decision trees. ETC like RF randomizes certain choices and subsets of information to play down over-learning from the information and over-fitting.

3.2.5 Decision Tree (DT)

The goal of employing a decision tree is to form an education model which can make use of to expect the lesson or esteem of the target variable by way of studying sincere choice policies deduced from in advance information (training facts). In choice bushes, for predicting a category name for a record we start from the root of the tree. We evaluate the values of the foundation property with the record's best. On the idea of comparison, we take after the branch comparing to that esteem and bounce to any other node.

$$E(S) = \sum_{i=1}^{C} -p_i p_i \quad (4)$$

In Equation 4, where S is a current state and the p_i is the probability of the ith state of S.

3.2.6 K. Neighbors (KNN)

K-nearest neighbor (KNN) can be an exceptionally basic, easy to get it, bendy and one of the topmost machine mastering algorithms. KNN is utilized in an assortment of applications inclusive of the fund, healthcare, political science, penmanship discovery, photograph acknowledgment

and video acknowledgment. In credit score value determinations, financial organizing will anticipate the credit rating of customers. In credit price, keeping money set up will anticipate whether the credit is comfortable or risky. In political technological know-how, classifying capacity voters in classes will vote or receive the vote. KNN calculation is applied for each type and relapse issue. Moreover, it is based totally on the spotlight likeness approach.

3.2.7 Stochastic Gradient Descent (SGD) Classifier

Utilizing stochastic gradient descent (SGD) on regularized direct approaches can aid in constructing an estimator for both classification and regression problems. The Scikit-learn API provides a module to implement the strategy specifically for classification issues. The SGD classifier employs a regularized linear model with the aim to build an estimator. It demonstrates robust performance with large-scale datasets and is an efficient and straightforward strategy to implement.

$$\theta_j = \theta_j - \alpha\ (\widehat{y^i} - y^j)x_j^i \tag{5}$$

In Equation 5 where θ_j estimator of the $(\widehat{y^i} - y^j)x_j^i$

3.2.8 Support Vector Classifier (SVC)

The goal of a linear SVC is too healthy to the data, returning a "find match" hyperplane that categorizes statistics. Based on this hyperplane, you may then feed some functions for your classifier to look at what the "anticipated" magnificence is.

3.2.9 Ada Boost Classifier

The Ada Boost algorithm, short for adaptive boosting, serves as a boosting technique employed in ML. As an ensemble approach, it earns its designation as adaptive boosting by reallocating weights to each instance, assigning higher weights to inaccurately labeled occurrences.

Boosting is utilized to decrease inclination as well as the fluctuation for administered learning. It works on the rule where learners are developed consecutively. However, for the primary level, each ensuing learner is developed from already developed learners. In straightforward words, frail learners are changed over into strong ones. Ada Boost calculation too works on the same guideline as boosting, however, there is a slight contrast in working.

4. Proposed Methodology

This research employs a variety of techniques to address classification challenges. The approach for resolving the classification problem is depicted in Fig.1. The initial stage involves the processing of information during the initialization phase. The study categorizes this information into six classes: joy, sadness, fear, hate, greed and anger as shown in Table 5.

During the testing phase, assessment texts are converted to lowercase letters as a preliminary step. Following this, the process involves applying segmentation to critiques and subsequently utilizing the stemming method on the reviews. This is done to achieve the fundamental structure of each phrase. As part of the preprocessing stage, certain phrases that tend to create confusion in text analysis are removed from the textual content reviews. The outcomes of this preprocessing are illustrated in Table 6, displaying the sample data after undergoing these steps.

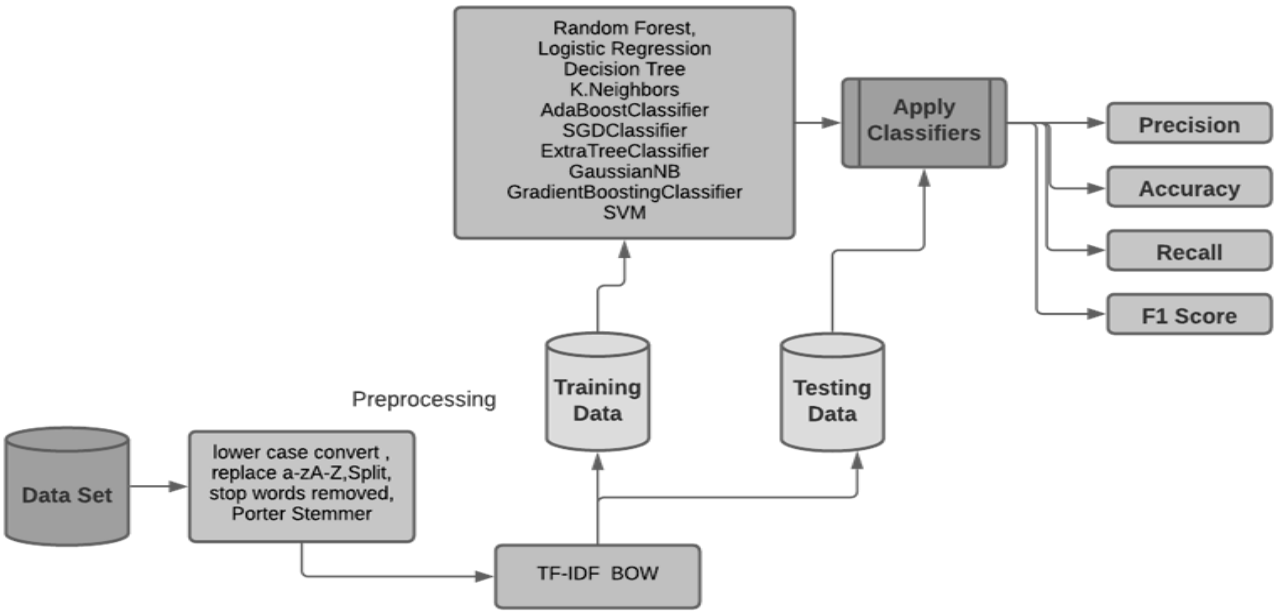


Fig.1 Proposed Methodology

Once the preprocessing is complete, the dataset is divided into subsets for both training and testing purposes. This division adheres to an 80-20 ratio, where 80% of the data is used for training and the remaining 20% is allocated for testing.

Table 5. Quantity of Samples corresponding to each target elegance

Tweets data	Target Value
I see the Tesla fan boys are angry at the idea	Anger
amritabithi You're gonna hate it when it is	Hate
This makes me very sad. Dwane was a great	Sad
The Fear and Greed Index big	Greed
The Fear and Greed Index big	Fear
+8.73% over the last hour JOY crypto	Joy

Table 6 Tweets Preprocessing

Without Preprocessing	With Preprocessing
I see the Tesla fanboys are angry at the idea	see tesla fanboy angry idea
amritabithi You're gonna hate it when it is	amritabithi go hate
This makes me very sad. Dwane was a great	Make very sad.d wane great
The Fear and Greed Index big	fear greed index big
The Fear and Greed Index big	fear greed index big
+8.73% over the last hour JOY crypto	+8.73%over last hour joy crypto

5. Evaluation

In this section, we engage in a discussion about solving grouping problems using various models. The utilization of TF-IDF and BoW techniques for feature selection is explored within the context of the article. The article employs TF-IDF and BoW methods, leading to more robust conclusions. A comparative analysis is tested empirically from models like RF, DT, SGD classifier, KNN, AC, GNB, GB classifier and ETC against the LR.

The time complexity of distinct ML models differs concerning both testing and training times. Although there's minimal variance in testing time among these models, training time displays significant fluctuations contingent on the complexity of the ML model. For instance, linear regression and Naive Bayes are simple models with linear time complexity, whereas SVM exhibits a complex nature with quadratic time complexity during the training phase. However, it's important to note that while our research focuses on accuracy and other pertinent parameters, the aspect of time complexity isn't directly intertwined with the primary contributions of our work.

Within our research, we deliberately select MLmodels with diverse parameters. These parameters are chosen through empirical methods to achieve optimal accuracy. The Gaussian Naive Bayes (GNB) classifier, operating on a likelihood basis, attains the lowest accuracy compared to other models. While KNN excels in accuracy for smaller datasets, its performance diminishes on our larger dataset. On the other hand, the amalgamation of linear models in random forest yields comparatively favorable results. Models such as ETC, DT and ADB exhibit enhanced accuracy, particularly for multi-class datasets, outperforming the GNB model. Notably, GB classifier, logistic regression, support vector machine (SVM) and SGD classifier exhibit the highest accuracy among the models employed in our research. This is due to the clarity and labeling present in our dataset, aligning with the conditions necessary for these models to excel. Conversely, should our dataset lack proper labeling or preprocessing, these models would perform sub optimally due to the potential noise present in the data. Given our dataset's probabilistic nature, the GNB model produces exceptional results across all models.

We empirically fine-tune these parameters to achieve peak accuracy. For instance, logistic regression, SGD classifier, SVM and GB classifier demonstrate superior accuracy, precision, recall and F1-score through the application of multinomial parameters. The outcomes of TF-IDF are presented in Table 7 to provide a comprehensive view of the results.

Table 7 Result in TF-IDF

Models	Accuracy	Precision	Recall	F1-score
RF	0.9	0.77	0.75	0.76
LR	0.93	0.84	0.8	0.82
GNB	0.36	0.36	0.53	0.33
ETC	0.87	0.72	0.72	0.72
DT	0.89	0.76	0.74	0.75
KNN	0.81	0.61	0.67	0.62
Stochastic Gradient Descent	0.93	0.86	0.79	0.84
SVM	0.93	0.84	0.8	0.81
Aboost	0.92	0.85	0.82	0.83
Gradient Boosting Classifier	0.93	0.80	0.80	0.81

In our investigation, the data utilized for this experimentation was sourced from Kaggle, a platform that hosts a substantial collection of opposing tweets. The data is stored in JSON format and consists of six distinct file types: joy.json, sadness.json, fear.json, hateful.json, anger.json and greed.json. While previous works on datasets lacked a diversity of classes conducive to aiding cryptocurrency miners in making more informed decisions, this study takes a different approach by categorizing data into six emotional classes. This classification empowers cryptocurrency minors to gain insights into people's emotions, thus enhancing their decision-making process.

Within this study, a variety of ML models with differing parameters were chosen. These parameters were meticulously fine-tuned through empirical methods to attain the highest possible levels of accuracy, precision, recall and F1-score. It's noteworthy that the GNB classifier yields the lowest values for accuracy, precision, recall and F1-score due to its likelihood-based operation, causing it to lag behind other models. Similarly, KNN performs exceptionally well with

accuracy, precision, recall and F1-score in small datasets, but its performance diminishes when applied to our larger dataset.

The random forest (RF) model, which amalgamates various linear models, produces moderately improved outcomes. Conversely, models such as: ET, DT and ADB exhibit improved accuracy, precision, recall and F1-score due to their compatibility with multi-class datasets similar to ours.

However, it is the GB classifier, LR, SVM and SGD classifier that deliver the most impressive accuracy, precision, recall and F1-score among all models employed in our research. This superiority is attributed to our dataset's clear labeling, aligning perfectly with the conditions required for these models to excel. In contrast, if our dataset lacks proper labeling or preprocessing, these models perform inadequately due to potential noise, leading to subpar accuracy, precision, recall and F1-score.

Given our dataset's probabilistic nature, the Gaussian Naive Bayes model excels across all models. Fig. 2 visually represent applying GMB give less than others models accuracy, precision, recall and F1-score with TF-IDF. Figures 3 and 4 visually represent the contrast between the least and greatest values for accuracy, precision, recall and F1-score. Fig. 3 illustrates GNB's least favorable performance using TF-IDF, while also showcasing LR, SGD classifier, GB and SVM models' optimal performance. Fig. 4 shows comparative analysis for different selected models with BOW.

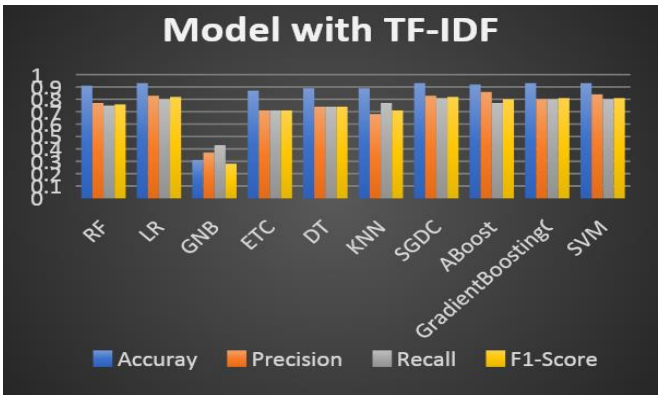


Fig. 2 Classifier Training-cross Validation Curve

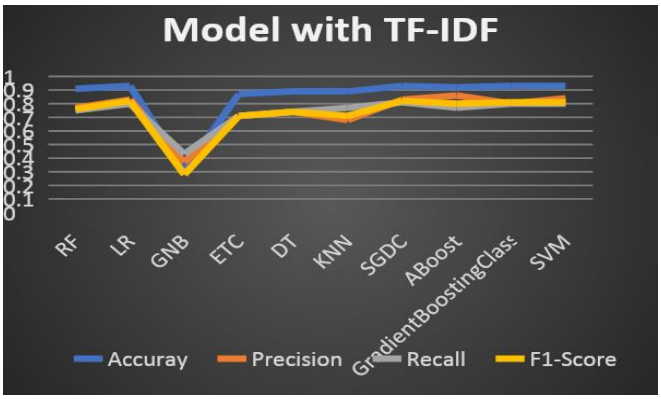


Fig. 3 Results of Models using TF-IDF

We determined these parameters empirically to attain optimal levels of accuracy, precision, recall and F1-score. As an illustration, the linear parameter yields the highest values for accuracy, precision, recall and F1-score among the SVM models. The outcomes for BoW are detailed in Table 8.

Table 8 Result in BoW

Models	Accuracy	Precision	Recall	F1-score
RF	0.91	0.77	0.75	0.76
LR	0.93	0.83	0.80	0.82
GNB	0.31	0.37	0.43	0.28
ETC	0.87	0.71	0.71	0.71
DT	0.89	0.74	0.74	0.74
KNN	0.89	0.68	0.77	0.71
Stochastic Gradient Descentc	0.93	0.83	0.81	0.82
Ada Boost	0.92	0.86	0.77	0.80
Gradient Boosting Classifier	0.93	0.80	0.80	0.81
SVM	0.93	0.84	0.8	0.81

Fig. 5 illustrates the minimum accuracy, precision, recall and F1-score achieved by GNB when utilizing BoW. Additionally, it displays the maximum accuracy, precision, recall and F1-score obtained by LR (Logistic Regression), SGD (Stochastic Gradient Descent), GB (Gradient Boosting) and SVM (Support Vector Machine).

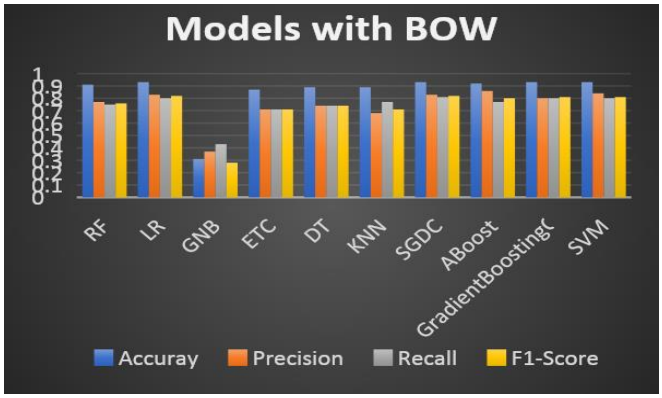


Fig. 4 Clustered Column Chart Showing Results of Models using BoW

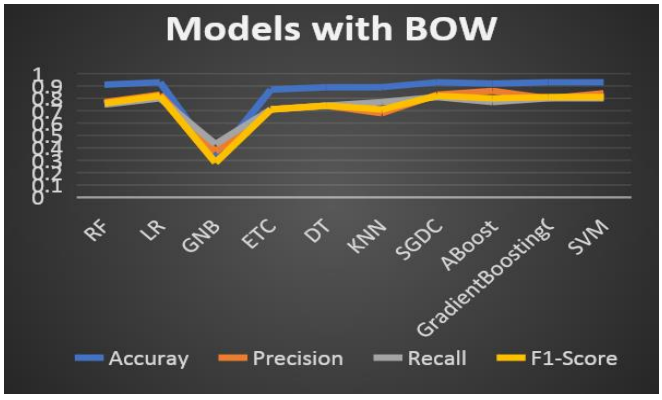


Fig. 5 Line Chart Showing the Results of Models using BoW

5. Conclusion

In this research, we harness a variety of ML techniques to address the task of categorizing user reviews. We employ feature engineering methods like TF-IDF and BoW to facilitate this classification process. A range of classifiers including: RF, LR, DT, KNN, AC, GNB, Extra Tree, SGD Classifier and SVM were trained on textual reviews. Their objective was to predict the emotional tone of user reviews, encompassing sentiments like sadness, joy, hate, anger, greed, and fear exclusively. Our findings underscore that the conclusions drawn from our tests hinge on a singular dataset that has not been previously employed for classification purposes. Additionally, the outcomes might be constrained to the specific dataset employed here. Among the models assessed, the top-performing quartet consists of SVM, LR, SGD, and GBC. These models exhibit superior accuracy, precision, recall, and F1-score metrics.

6. Future Work

Subsequent endeavors will encompass the utilization of deep learning models for conducting tests across diverse textual and categorical datasets aimed at categorizing user reviews.

References

[1] D. Garcia, C. J. Tessone, P. Mavrodiev & N. Perony, "*The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy"*, Journal of the Royal Society Interface, vol. 11, 2014.

[2] J. Abraham, D. Higdon, J. Nelson and J. Ibarra, "*Cryptocurrency price prediction using tweet volumes and sentiment analysis"*, SMU Data Science Review, vol. 1, no.3, 2018.

[3] D. Kahneman and A. Tversky, "*Prospect theory: An analysis of decision under risk*", *Handbook of the fundamentals of financial decision making: Part I*., World Scientific. pp. 99-127, 2013.

[4] F.F. Bocca and L.H.A. Rodrigues, "*The effect of tuning, feature engineering, and feature selection in data mining applied to rainfed sugarcane yield modelling"*, Computers and electronics in agriculture, no. 128, pp. 67-76, 2016.

[5] A. Jafar, M.A. Islam, and A. Khan, "*The Effect of Social Networks and Google on Consumers' Buying Behavior in Dhaka City, Bangladesh"*, Global Journal of Management and Business Research, no. 17, 2017.

[6] M. Keijsers, C. Bartneck, and H.S. Kazmi, "*Cloud-based sentiment analysis for interactive agents*", in *Proceedings of the 7th International Conference on Human-Agent Interaction*, 2019.

[7] T.H. Nguyen, K. Shirai, and J. Velcin, "*Sentiment analysis on social media for stock movement prediction"*, Expert Systems with Applications, vol. 42, pp. 9603-9611, 2015.

[8] H. Sul, A.R. Dennis, and L.I. Yuan. "*Trading on twitter: The financial information content of emotion in social media*", *47th Hawaii International Conference on System Sciences*, IEEE, 2014.

[9] A. Mittal and A. Goel, "*Stock prediction using twitter sentiment analysis"*, Standford University, CS229, vol. 15, pp. 2352, 2012.

[10] J. Bollen, H. Mao and X. Zeng, "*Twitter mood predicts the stock market"*, Journal of computational science, vol. 2, no. 1, pp. 1-8, 2011.

[11] S.B. Kotsiantis, I. Zaharakis, and P. Pintelas, "*Supervised machine learning: A review of classification techniques.* Emerging artificial intelligence applications in computer engineering", vol. 160, no. 1, pp. 3-24, 2007.

[12] C. Lamon, E. Nielsen, and E. Redondo, "*Cryptocurrency price prediction using news and social media sentiment"*, SMU Data Sci. Rev, vol. 1, no. 3 pp. 1-22, 2017.

[13] Z.H. Munim, M.H. Shakil, and I. Alon, "*Next-day bitcoin price forecast"*, Journal of Risk and Financial Management, vol. 12, pp. 103, 2019.

[14] B. M. Lucey and M. Dowling, "*The role of feelings in investor decision-making"*, Journal of economic surveys, vol. 19, no. 2, pp.211-237, 2005.

[15] M.M. Seif, E.M. Ramzy, and G. Abdel, "*Stock market real time recommender model using apache spark framework*", *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, Springer, 2018.

[16] N. Aslam, F. Rustam, E. Lee, P.B. Washington and I. Ashraf, "*Sentiment analysis and emotion detection on cryptocurrency related tweets using ensemble LSTM-GRU model"*, IEEE Access, vol. 10, pp. 39313-39324, 2022.

[17] J. Heaton, "*An empirical analysis of feature engineering for predictive modeling*", in *SoutheastCon 2016*, IEEE, 2016.

[18] S. Robertson, "*Understanding inverse document frequency: on theoretical arguments for IDF"*, Journal of documentation, vol. 60, no. 5, pp. 503-520, 2004.

[19] F. Rustam, I. Ashraf, A. Mehmood, S. Ullah, and G.S. Choi, "*Tweets classification on the base of sentiments for US airline companies"*, Entropy, vol. 21, no. 11, pp. 1078, 2019.

[20] D. Shah, H. Isah, and F. Zulkernine, "*Predicting the effects of news sentiments on the stock market*", *IEEE International Conference on Big Data (Big Data)*, IEEE, 2018.

[21] B. Spilak, "*Deep neural networks for cryptocurrencies price prediction*", Humboldt-Universität zu Berlin, 2018.

[22] E. Stenqvist and J. Lönnö, "*Predicting Bitcoin price fluctuation with Twitter sentiment analysis"*, 2017.

[23] P.C. Tetlock, "*Giving content to investor sentiment: The role of media in the stock market"*, The Journal of Finance, vol. 62, no. 3, pp. 1139-1168, 2007.

The Nucleus ***61***, *No. 1 (2024) 10-15*

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

On the Elliptical Orbit of the Earth and Position of the Sun in the Sky: An Engineering Approach

Ruben Avila and S. Shoaib Raza*

Department of Thermo-fluids, Faculty of Engineering, National Autonomous University of Mexico (UNAM), Mexico D.F., C.P. 04510, Mexico

ABSTRACT

The position of the Sun as seen by an observer on the Earth's surface and the position and velocity vectors of the Earth revolving in an elliptical orbit around the Sun can be calculated using several computational approaches. These approaches include (but are not limited to) the use of an analytical approach; a numerical approach, and the use of a Solar Position Algorithm (PSA). In the analytical methodology, the Earth's momentum equation is transformed to eliminate its time dependence, and the equation is solved analytically. Whereas, using the numerical approach, the dimensionless momentum equation of the revolving Earth is written in the polar coordinate system (r, θ) and solved numerically. The solar position algorithm known as PSA (Plataforma Solar de Almeria, abbreviated from its Spanish origin: https://www.psa.es), is a numerical algorithm that uses several empirical relations to calculate the solar declination and the ecliptic longitude angles, etc. The algorithm uses Cartesian coordinate system to calculate the dimensionless coordinates of the pole star (Polaris) and its declination angle to calculate the position vector of an observer that rotates with the Earth. This coordinate system is referred to as a new Cartesian coordinate system whose origin is located at the center of the Earth. The solar elevation angle and azimuth angle are obtained by performing a set of rotations of this new Cartesian coordinate system. In this article, we have used basic physical principles (analytical approach) to obtain the main parameters of the Sun's trajectory and position, at certain time in the sky. The methodology presented here can easily be used by professionals and engineers working in the area of solar/alternate energy, as well as for the design of intelligent/green buildings/cities for a sustainable environment.

Keywords: *Position of Sun; Solar Trajectory; PSA; Declination Angle; Orbit of Earth*

1. Introduction

The variation in the position of the Sun in the sky over an observer is a natural phenomenon that has intrigued humankind forever. The position of the Sun has been correlated with the occurrence of natural phenomena (volcanic activity, storm cycles, and earthquakes). The motion of the Sun has also been considered as a measure of time, or as a phenomenon that governs the agricultural cycles and diseases. Since the 20th century, the accurate determination of the Sun's position has been an important subject of study for engineers. Due to the increasing price of petroleum, the effect of greenhouse gases on climate change and global warming, and the increasing number of internal combustion engines in big cities, it is necessary that engineers develop alternative energy sources. Engineers need to efficiently extract energy from renewable and free sources, such as the Sun. Energy engineers need to design efficient solar furnaces, solar steam generators, solar water heaters, solar cells, etc. If civil engineers efficiently use solar energy, they may design reliable intelligent buildings and sustainable environments. In the near future, the task of the engineers will be very important. However, sometimes they do not have enough background to understand the mathematical notations that the physicists and astronomers use to calculate the Sun's position. In view of the above, we have made a literature review and explained the methodology, here.

The determination of the Sun's position in the sky by using vector analysis techniques has been previously reported by [1] and [2]. A simple parametric model, that describes the basic principles of the visible Sun's path on the celestial sphere, has been presented by [3]. A review of the Sun's position algorithms that were published in the solar literature is presented by [4]. The Sun position algorithms are sophisticated schemes, that compute the position of the Sun in the ecliptic, celestial, and horizontal coordinates, see [5]. Very recently, a review of the Sun position algorithms has been presented in [6]. On the internet sites, it is also possible to find and execute computer codes to calculate the position of the Sun in the sky, for instance [7]. The purpose of this paper is to present a self-contained material suitable for energy and civil engineers/researchers to determine the solar position in the sky.

The paper is organized as follows. In section 2, the Earth's orbit equation is presented. In section 3, the methodology to obtain the Cartesian coordinates of the star Polaris and the calculation of the declination angle, are presented. In section 4, a Cartesian coordinate system, whose origin is located at the center of the Earth, is introduced to define both the position vector of an observer and the position vector of the Sun. Employing a set of rotations, the solar elevation angle and the solar azimuth angle measured from the north are calculated. Discussion on results and Conclusions are presented in sections 5 and 6, respectively.

2. Earth's orbit equation

In the mathematical model of the Earth's orbit equation, it is assumed that the Earth is being attracted to a fixed attracting focus (the Sun). The motion is confined to the ecliptic plane, which is described by the radius vector (from the Sun to the Earth) and the velocity vector of the Earth. Using a polar coordinate system r - θ, where r (the radial coordinate) is measured from a fixed focus (the Sun) and θ (the angular coordinate) is measured from a fixed reference line (the line traced from the Sun to the Earth at the Perihelion position), see Fig. 1.

*Corresponding author: ssraza@msn.com

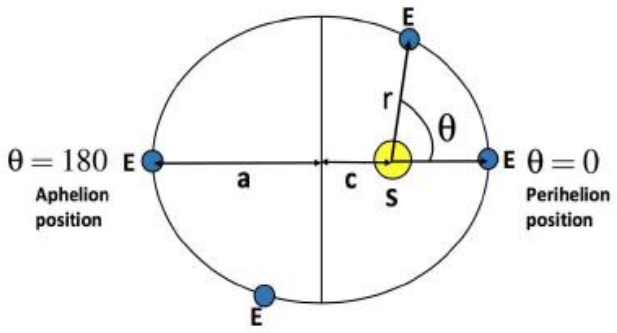


Fig. 1. Polar r - θ coordinate system. The letters S and E correspond to the Sun and the Earth positions, respectively. a and c represent the major axis and the focus of the elliptical trajectory of the Earth around the Sun, respectively. The Earth's Perihelion position (θ=0º) and the Earth's Aphelion position (θ=180º) are also shown.

The radial and angular components of the momentum equation of the Earth (in terms of force and acceleration) are written as (in the model it is assumed that the only force acting on the Earth is in the negative radial direction).

Radial component

If we assume that the motion of the Earth relative to the Sun, is the same as if the Sun is fixed and the mass of the Earth is replaced by the reduced mass M'_E, which is defined as

$$M'_E = \frac{M_E M_S}{M_E + M_S} \tag{1}$$

where M_S is the mass of the Sun and M_E is the mass of the Earth and the gravitational force in the radial direction (attracting force) is balanced by the centripetal force, we get (G is the gravitational constant):

$$\ddot{r} - r\dot{\theta}^2 = -\frac{G\,(M_E + M_S)}{r^2} \tag{2}$$

or

$$\ddot{r} - r\dot{\theta}^2 = -\frac{\mu}{r^2} \tag{3}$$

Where μ is the gravitational coefficient (positive constant) given as: $\mu = G\,(M_E + M_S)$. Please note dot (.) represents a time derivative and r and θ are shown in Fig.1.

Angular component

Considering that angular momentum is zero, the angular momentum component per unit mass h, is independent of time and is defined as:

$$h = r^2\dot{\theta} \tag{4a}$$

or (squaring both sides and rearranging)

$$\dot{\theta}^2 = \frac{h^2}{r^4} \tag{4b}$$

The principle of conservation of angular momentum states that the moment of the total external force applied to the Earth is equal to the time rate of change of the angular momentum of the Earth about the Sun. If the external moment is equal to zero, the angular momentum h must be a constant. Substituting Eq. (4b) into Eq. (3), we obtain

$$\ddot{r} - r\frac{h^2}{r^4} = -\frac{\mu}{r^2} \tag{5}$$

or

$$\frac{d^2r}{dt^2} = \frac{1}{r^2}\left(\frac{h^2}{r} - \mu\right) \tag{6}$$

2.1 Solution of the Earth's orbit equation: A numerical approach

Using the non-dimensional variables r* = r/a (where a is the semi-major axis of the Earth's elliptical orbit) and t* = tμ/(ah), the dimensionless radial component of the momentum equation is written as

$$\frac{d^2r^*}{dt^{*2}} = \left(\frac{h^4}{a^2\mu^2 r^{*3}} - \frac{h^2}{a\mu r^{*2}}\right) \tag{7}$$

For the numerical solution of Eq. (7), we have considered the following values (which are available in the literature) of the Earth trajectory: (i) the semi-major axis a = 150 × 10^9 m, (ii) the angular momentum per unit mass h = 4,452,990,073 km^2/s [8] and (iii) the gravitational coefficient μ is obtained by employing the values of G = 6.673x10^{-11} $m^3/(kg\,s^2)$, M_E = 5.972 x 10^{24} kg and M_S = 2 x 10^{30} kg, as $\mu = G\,(M_E + M_S)$ = 132,774,392,455,423,200,000.0 m^3/s^2.

By substituting these values in Eq. (7), we have

$$\frac{d^2r^*}{dt^{*2}} = \frac{0.999442337}{r^{*3}} - \frac{0.99972113}{r^{*2}} \tag{8}$$

This second-order ordinary differential equation is solved by using a Runge-Kutta-Nystrom technique (D02LAF-NAG) and taking into account the following initial conditions: (i) at t*=0 the Earth's radial velocity is zero, that is, dr*/dt* = $\dot{r}^*$ = 0 and (ii) at t*=0, the Earth is at the Perihelion position i.e. θ = 0º. The numerical solution is performed along the whole year (365 days), by considering 525,600 time steps, which corresponds to a dimensionless time increment Δt*=1.19023 x 10^{-5}, which is equivalent to a dimensional time increment Δt=60s. By solving Eq. (8), we obtain, as a function of time t*, the dimensionless radial position (r*) and the dimensionless radial component of the velocity vector (dr*/dt* = $\dot{r}^*$). The dimensional tangential velocity of the Earth ($v_t = r\dot{\theta}$), which is called the orbital speed, is obtained from Eq. (4a), which says $v_t = r\dot{\theta} = h/r$.

The angle θ, around the Sun is calculated from the equation of an ellipse, which in polar coordinates seems as:

$$r = \frac{a(1-\varepsilon^2)}{1+\varepsilon\cos\theta} = \frac{l}{1+\varepsilon\cos\theta} \tag{9}$$

Where, ε is the eccentricity of the Earth's orbit, currently ε ≈ 0.0167005 and l is the semi-latus rectum defined as $l = a(1-\varepsilon^2)$ = 149,958,163,991.09 m. The dimensionless version of Eq. (9) is

$$r^* = \frac{(1-\varepsilon^2)}{1+\varepsilon\cos\theta} = \frac{l^*}{1+\varepsilon\cos\theta} \tag{10}$$

where $l^* = (1-\varepsilon^2) = 0.9997210932$. Hence the angle θ is obtained by rearranging the above equation, as:

$$\theta = \cos^{-1}\left(\frac{l^*-r^*}{r^*\varepsilon}\right) \quad (11)$$

From Eq. (11) it is clear that when θ=0° (defined as the Perihelion position, around January 3rd), the Earth is closest to the Sun, at a dimensional radial distance r equal to r_p, as:

$$r_p = a(1-\varepsilon) \approx 147.5 \times 10^6 \text{ km} \quad (12)$$

while when θ =180° (defined as the Aphelion position, around July 4th), the Earth is farthest from the Sun, at a dimensional radial distance, r equal to r_A, as:

$$r_A = a(1+\varepsilon) \approx 152.5 \times 10^6 \text{ km} \quad (13)$$

Then, the dimensionless polar coordinates at the Perihelion r^*_P and Aphelion r^*_A positions (from Eq. 10) are $r^*_P = (1-\varepsilon) = 0.9833$ and $r^*_A = (1+\varepsilon) = 1.0167$, respectively. From the numerical solution of Eq. (8), we obtain $r^*_{Pn}=0.9833$ (when θ=0°), $r^*_{An}=1.016699$ (when θ=180°) and $l^*_n=0.999721$, where, the subscript n refers to the use of the numerical approach.

2.2 Solution of the Earth's orbit equation: An analytical approach

If the above equations are reformulated to eliminate time dependence, the time derivatives of the radial distance, *r* are eliminated, and after applying considerable mathematics (for further details of the procedure please contact the corresponding author) and the use of available data for μ, *h* and *a*, the dimensionless Earth's elliptical orbit equation is obtained as:

$$r^* = \frac{r}{a}$$

$$= 0.99972109732795145 / (1+0.016700379398341621 \cos\theta) \quad (14)$$

If Kepler's second law of planetary motion that states "the radius vector from planet to Sun, sweeps equal areas in equal times as the planet orbits the Sun", we obtain an expression that relates the Earth's angle θ around the Sun to the elapsed time since $\theta = 0$ radians (that is, angle from the Earth's perihelion position). If the small element of the area in the elliptical Earth's orbit consists of a small isosceles triangle whose sides have length *r* and whose base length is $rd\theta$. The small area is given as:

$$dA = \frac{1}{2}r^2 d\theta \quad (15)$$

In a short time *dt* the Earth has the constant areal velocity, given by:

$$\frac{dA}{dt} = \frac{1}{2}r^2\frac{d\theta}{dt} \quad (16)$$

After applying a considerable mathematics, using Eq. (3) and the principle of conservation of angular momentum, we obtain:

$$dA = \frac{1}{2}\left[\frac{(1+\varepsilon)^2 r_p^2}{(1+\varepsilon\cos\theta)^2}\right] d\theta \quad (17)$$

and,

$$\frac{1}{2}\left[\frac{(1+\varepsilon)^2 r_p^2}{(1+\varepsilon\cos\theta)^2}\right] d\theta = \frac{1}{2}hdt \quad (18)$$

which can be rewritten (after performing the integration over time) as:

$$t = \frac{(1+\varepsilon)^2 r_p^2}{h}\int_0^{2\pi}\frac{d\theta}{(1+\varepsilon\cos\theta)^2} \quad (19)$$

The dimensional analytical solution of Eq. (19) is

$$t = \frac{(1+\varepsilon)^2 r_p^2}{h}\left[\frac{2}{\sqrt{1-\varepsilon^2}}\tan^{-1}\left(\frac{\sqrt{1-\varepsilon^2}}{1+\varepsilon}\tan\left(\frac{\theta}{2}\right)\right) - \frac{\varepsilon\sin\theta}{1+\varepsilon\cos\theta}\right] \quad (20)$$

where the dimensional time, t, is in seconds. In the derivation of the above equations, the values for h=4,456,990,073,000,000 m^2/s and a=149,597,885,651 m, have been used. The dimensionless expression for the time t^* is given as:

$$t^* = \left(\frac{\mu}{ah}\right)\frac{(1+\varepsilon)^2 r^2_p}{h}\left[\frac{2}{\sqrt{1-\varepsilon^2}}\tan^{-1}\left(\frac{\sqrt{1-\varepsilon^2}}{1+\varepsilon}\tan\left(\frac{\theta}{2}\right)\right) - \frac{\varepsilon\sin\theta}{1+\varepsilon\cos\theta}\right] \quad (21)$$

From Eqs. (14) and (21), the values of r* and t* respectively, are calculated for a set of θ angles in the interval $0 \le \theta \le 2\pi$.

2.3 Solar position algorithm (PSA)

The Sun's position algorithm, namely, "Plataforma Solar de Almeria" (PSA algorithm - abbreviated from its Spanish origin: https://www.psa.es) developed by [4] is a numerical algorithm used to calculate, as a function of time (specified by the Julian day, the calendar date and the universal time), the Sun position parameters, such as the ecliptic longitude angle $\hat{\theta}$ (which is related with the polar coordinate, θ, see the previous sections) and the declination angle, β, among others. Following [4] the difference n, between the current Julian day (*jd*) and Julian day 2,451,545 (which corresponds to the day starting at 12:00 UT on January 1, 2000) is given by:

$$n = jd - 2451545 \quad (22)$$

Where, the current Julian day is obtained from:

$$jd = \frac{1461*(year+4800+jm1412)}{4} + \frac{367*(month-2-12*jm1412)}{12} - \left(\frac{3*(year+4900+jm1412)}{100*4}\right) + day - 32075 - 0.5 + \frac{h\hat{o}ur}{24} \quad (23)$$

Where, the parameter hôur includes the hour of the day (hour) in Universal Time and in decimal format (that is, the minutes and seconds as a fraction of an hour are also included), then

$$h\hat{o}ur = hour + \left[minutes + \frac{seconds}{60}\right]/60 \quad (24)$$

And, *jm*1412 = (*month*-14)/12. All divisions except the last one are integer divisions, see [4]. The ecliptic coordinates (the coordinates evaluated on the ecliptic plane) are computed for the required Julian day.

3. Location of the star Polaris (North Star) and the declination angle

To calculate the position of the Sun in the sky of an observer, we direct the Earth's rotation axis to the star Polaris. The coordinates of the North star are defined in a Cartesian coordinate system whose origin is located at the center of the Earth's elliptical orbit and whose plane x_1-x_2 defines the ecliptic plane, see Fig. 2. Where the subscript E, refers to the Earth; N to the North Star and S refers to the position of Sun.

We have assumed (considering that the North Star is far away from the Sun-Earth system) that the angle between the

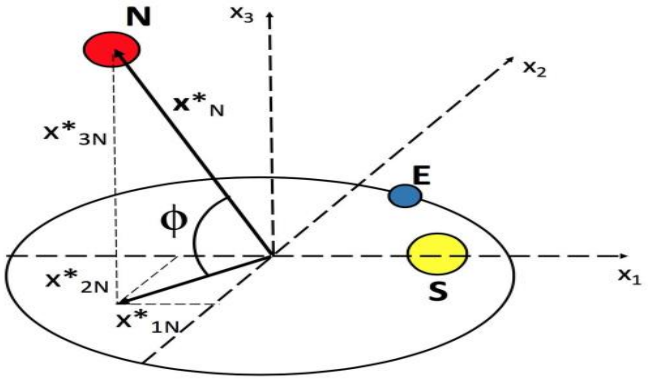


Fig. 2. Cartesian coordinate system, whose origin is located at the center of the Earth's elliptical orbit. In the figure positions of the Sun, Earth and the North Star (Polaris) are shown. Also shown is the angle (66.5477º) between the position vector of Polaris and the ecliptic plane (plane x_1−x_2 of the Cartesian coordinate system).

rotation axis of the Earth to the position vector of the North Star is the same as that of its position vector with the ecliptic plane (also see Fig. 3, below).

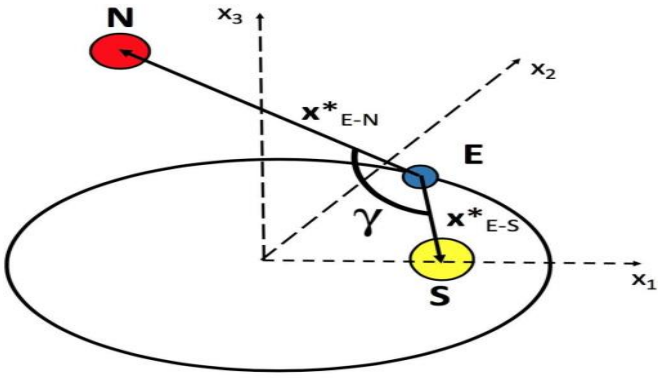


Fig. 3. The angle γ, between the Earth's rotation vector and the vector from the Earth to the Sun.

The angle γ, between the Earth's rotation vector x^*_{E-N} and the vector from the Earth to the Sun $x^*_{E-S} = x^*_S - x^*_E$, is obtained as (see also Fig. 4, below). Its components are:

$$x^*_{1E-S} = -r^* cos\theta, \;\; x^*_{2E-S} = -r^* sin\theta, \;\; x^*_{3E-S} = 0 \quad (25)$$

$$\gamma(t) = \cos^{-1}\left(\frac{x^*_{E-S}.\, x^*_{E-N}}{|x^*_{E-S}||x^*_{E-N}|}\right) \quad (26)$$

The declination angle β (t) between the Earth's equator and the vector from the Earth to the Sun x^*_{E-S} is given as (see Fig. 4, below).

$$\beta(t) = 90^o - \gamma(t) \quad (27)$$

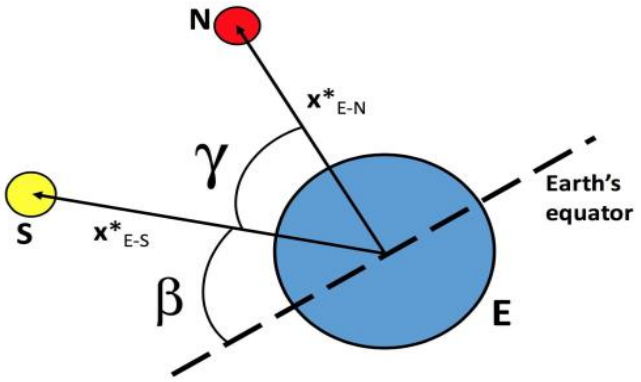


Fig. 4. The declination angle β between the Earth's equator and the vector from the Earth to the Sun x^*_{E-S}.

By performing a trial and error procedure, the North Star is calculated to be at the dimensionless coordinates $x^*_{1N} = 4 * 10^9$ and $x^*_{2N} = -1 * 10^9$

The third component of the position vector of the North Star is calculated as:

$$x^*_{3N} = (\, x^{*2}_{1N} + \, x^{*2}_{2N})^{\frac{1}{2}} \tan(\phi) = 9.5 * 10^9 \quad (28)$$

The convergence criteria of the trial and error process are based on the successful evaluation of the dates at which the equinoxes and solstices occur.

Fig. 5 shows the declination angle β and the ecliptic longitude $\hat{\theta}$ as functions of the days along the year. In the numerical solution (left panel), β is calculated from Fig. 4, while θ is calculated from Eq. (11) (and converted to the ecliptic longitude $\hat{\theta}$). In the analytical solution (middle panel), θ is the independent variable of Eq. (20) (and it is converted to the ecliptic longitude $\hat{\theta}$). In the Sun position algorithm (right panel), PSA is used to calculate $\hat{\theta}$. The calculations have been made for the year 2013. Note that for the three approaches, at the equinoxes, the declination angle β is equal to zero hence at the Earth's equator a vertical zenith is reached. While at the summer and winter solstices, the vector x^*_{E-S} passes through the Tropic of Cancer (i.e. β = 23.45º) and Tropic of Capricorn (i.e. β = - 23.45º) respectively.

The dimensionless tangential velocity (orbital speed) of the Earth that is calculated by the numerical and analytical algorithms is depicted in Fig. 6.

4. The position vector of an observer on Earth and the Earth's rotation

In order to consider the two motions of the Earth: (i) rotation about its axis that points towards the North star and (ii) the elliptical trajectory around the Sun, a new fixed Cartesian coordinate system (o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$) is defined, see Fig.7.

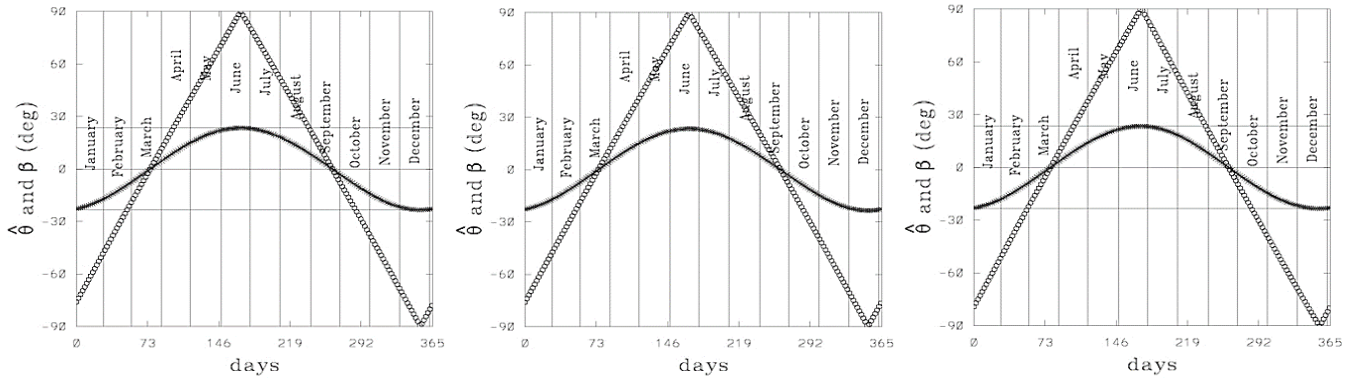


Fig. 5. The declination angle β (*) and the ecliptic longitude $\hat{\theta}$ (◦) as functions of time (days along the year). Left panel: Numerical solution, β is calculated from Fig. (4), while θ is calculated from Eq. (11) (and converted to the ecliptic longitude $\hat{\theta}$). Middle panel: Analytical solution: θ is the independent variable of Eq. (20) (and it is converted to the ecliptic longitude $\hat{\theta}$). Right panel: Sun position algorithm (PSA) is used to calculate $\hat{\theta}$.

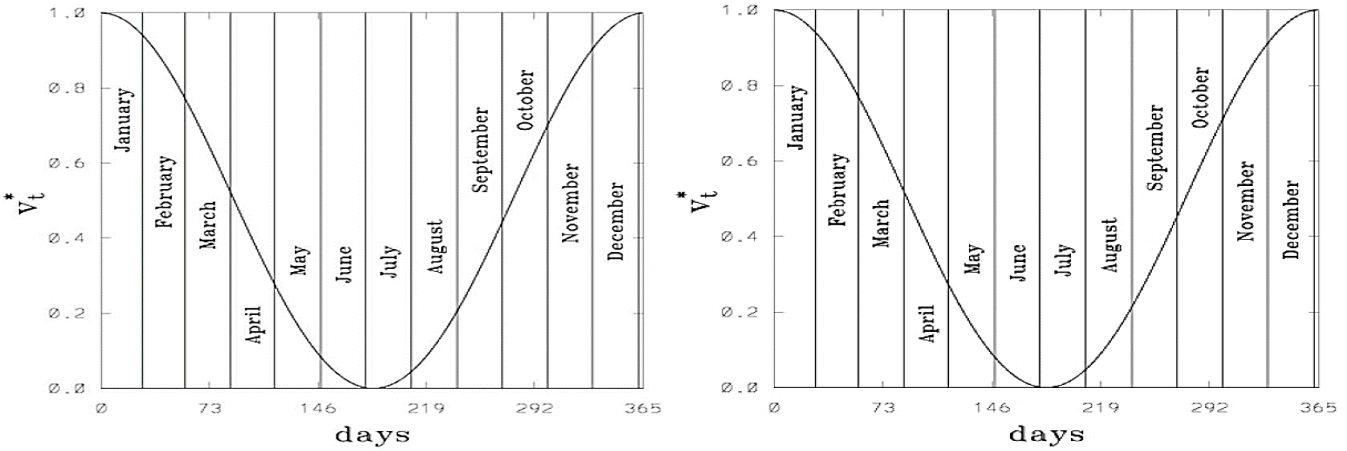


Fig. 6. Dimensionless tangential velocity (orbital speed) of the Earth, $v^*_t = (v_t - v_{tmin})/(v_{tmax} - v_{tmin})$ as a function of time (days along the year). The dimensional tangential velocity is calculated from Eq. (4), as $v_t = r\dot{\theta}$). Left panel: Numerical solution, r (which is the dimensional value of r*) is calculated from Eq. (8) (v_{tmax} = 30252.76 m/s and v_{tmin}= 29258.89 m/s). right panel: Analytical solution, r (which is the dimensional value of the variable r* = r/a, where a is the semi-major axis of the Earth's elliptical orbit) is calculated from Eq. (14) (v_{tmax}= 30299.14 m/s and v_{tmin}= 29303.75 m/s).

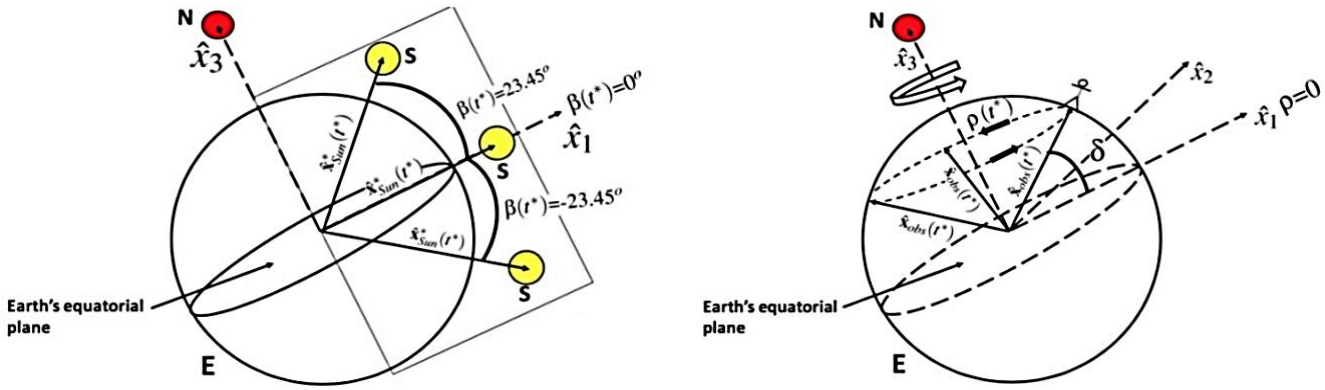


Fig.7. Cartesian coordinate system o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$ whose origin is located at the center of the Earth. Its $\hat{x}_3$ axis points towards the star Polaris and its plane $\hat{x}_1$ - $\hat{x}_2$ is on the Earth's equatorial plane. Left panel shows that the position vector of the Sun $\hat{x}*\text{Sun}(t^*)$ moves on the plane $\hat{x}_1$ - $\hat{x}_3$. Right panel shows the position vector of an observer $\hat{x}_{obs}(t^*)$, that is located at a certain fixed latitude δ on the Earth's surface and the rotation angle ρ.

It may be noted that $t^* = 0$ at $\rho = 0$

This coordinate system has the following characteristics: (i) its origin is located at the center of the Earth, (ii) its plane $\hat{x}_1$ - $\hat{x}_2$ is on the Earth's equatorial plane, (iii) the position vector of the Sun moves on the plane $\hat{x}_1$ - $\hat{x}_3$, (iv) its $\hat{x}_3$ axis points towards the star Polaris and (v) the orientation of its $\hat{x}_1$ axis is defined together with the initial value (at $t^* = 0$) of the rotation angle ρ, we have assumed that at $t^* = 0$, $\rho = 0$ radian. In this new Cartesian coordinate system, we define two vectors, the vector $\hat{x}_{obs}(t^*)$, which is the position vector of an observer that is located at a certain fixed latitude δ on the Earth's surface, and the vector $\hat{x}^*_{sun}(t^*)$, which is the Sun's position vector. Notice that the vector $\hat{x}_{obs}(t^*)$ rotates at the same angular velocity as the Earth, see right panel of Fig. 7. In the model, it is assumed that the Earth's rotation angle ρ is $0 \leq \rho \leq 2\pi$, where 2π radian, corresponds to 1 day (24 hours or 86400 seconds). The increment of the rotation angle Δρ (which corresponds to the time step Δt=60 s of the numerical solution) is calculated as:

$$\rho\Delta = \frac{2\pi * 60}{24 * 3600} = 0.00436\, radians \qquad (29)$$

The dimensionless three components of the rotating vector $\hat{x}^*_{obs}(t^*)$ referred to as the fixed Cartesian coordinate system are given as:

$$\hat{x}^*_{1obs}(t^*) = \cos\delta\cos\rho(t^*),$$
$$\hat{x}^*_{2obs}(t^*) = \cos\delta\sin\rho(t^*), \hat{x}^*_{3obs}(t^*) = \sin\delta \qquad (30)$$

while the dimensionless three components of the Sun's position vector $\hat{x}^*_{1sun}(t^*)$, which oscillates from $\beta(t^*) = -23.45^o$ to $\beta(t^*) = 23.45^o$ on the plane $\hat{x}_1$ - $\hat{x}_3$, are the following:

$$\hat{x}^*_{1sun}(t^*) = cos\beta(t^*), \hat{x}^*_{2sun}(t^*) = 0, \hat{x}^*_{3sun}(t^*) = sin\beta(t^*)$$

It may be noted that in Eq. (30), the dimensionless radius of the Earth is taken as equal to 1.

5. Results and Discussion

Some of the preliminary results obtained using these computational methodologies viz., the Numerical approach, Analytical approach, and the PSA are presented in Figures 5 and 6. Fig. 5 shows the declination angle β and the ecliptic longitude $\hat{\theta}$ as functions of the 365 days along the year 2013 (chosen arbitrarily to demonstrate the methodology). Both the numerical and analytical solutions compare well with the PSA which demonstrates the suitability and validity of our computational methodology. It may be noted in Fig. 5, that at the time of equinoxes, the declination angle β is equal to zero hence at the Earth's equator, a vertical zenith is reached. While at the summer and winter solstices, the vector x^*_{E-S} passes through the Tropic of Cancer (i.e. $\beta = 23.45^o$) and Tropic of Capricorn (i.e. $\beta = -23.45^o$), respectively.

The tangential velocity (orbital speed) of the Earth is calculated by the numerical and analytical algorithms. It can be observed in the figure (see Fig. 6) that the dimensionless tangential velocity of Earth predicted by both the methodologies compare well, hence elaborating the success of our computations.

6. Conclusions

Diverse computational methodologies have been presented to calculate the trajectory of the Sun in the sky of an observer located on the Earth's surface. A numerical algorithm and an analytical methodology have been used to get the parameters needed to obtain the Sun's position in our sky. The location of the North Star has been calculated in a Cartesian coordinate system, which is a familiar coordinate system for engineers. Additionally, for the calculation of the location of the North Star, the position vector from the Earth to the Sun γ and the declination angle β as a function of time, were obtained for the use of the energy engineers. Standard transformations of the involved vectors (the position vector of the observer and the Sun's position vector) have been obtained by performing simple rotations of the Cartesian coordinate system.

The information included in this paper, although is a standard one, should be considered as an important source of reference, for solar energy engineers/civil engineers. For the construction of intelligent buildings for a sustainable environment, engineers may use this approach to accurately know the position of the Sun in the sky throughout the year.

Acknowledgment:

The authors acknowledge the support of the DGAPA-PAPIIT project IN117314. The computations presented in this paper were performed on the workstations acquired through the funds provided by this project.

References

[1] A.B. Sproul, "Derivation of the solar geometric relationships using vector analysis," Renewable Energy, Vol. 32, pp. 1187–1205, 2007.

[2] A. Jenkins, "The Sun's position in the sky," Eur. J. Phys., Vol. 34, pp. 633–652, 2013.

[3] V. Khavrus and I. Shelevytsky, "Introduction to solar motion geometry based on a simple model," Phys. Ed., Vol. 45, pp. 641–653, 2010.

[4] M. Blanco-Muriel, D.C. Alarc´on-Padilla, T. Lopez-Moratalla, and M. Lara-Coira, "Computing the solar vector," Solar Energy, Vol. 70, pp. 431–441, 2001.

[5] I. Reda and A. Andreas, "Solar position algorithms for solar radiation applications" Tech. Rep. NREL/TP-560-34302, National Renewable Energy Laboratory, Golden, Colorado, USA, January 2008.

[6] G. Prinsloo and R. Dobson, Solar Tracking. Stellenbosch: Solar Books, 2015.

[7] National Oceanic and Atmospheric Administration, US Department of Commerce, "NOAA Solar Calculator," assessed from the website of esrl: https://gml.noaa.gov/grad/solcalc/

The Nucleus ***61***, *No. 1 (2024) 16-21*

www.thenucleuspak.org.pk

The Nucleus
ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Role of Neutron Beam Applications in the Sustainable Socio-Economic Development of Pakistan

Khurram Shahzad* and M. Nasir Khan

Pakistan Nuclear Society (PNS), Islamabad, Pakistan

ABSTRACT

Pakistan Institute of Nuclear Science and Technology (PINSTECH) is the largest multidisciplinary research and development center in Pakistan. It has two research reactors namely PARR-1 and PARR-2. Neutron beam facilities have been installed only around PARR-1. It is a multipurpose research reactor that has been mostly utilized for the production of radioisotope and the training of scientists, engineers and technicians. The reactor is also utilized in the studies of neutron reaction cross-sections, nuclear structure, fission physics, radiation damage in crystals and semi-conductors, studies of geological, biological and environmental samples by neutron activation techniques, gemstone coloration, neutron radiography and neutron scattering. We have been investigating lattice dynamics, crystal and magnetic structure of materials and residual stress measurement using neutron scattering. In this manner, PARR-1 is effectively contributing to the socio-economic development of Pakistan.

Keywords: *Research reactors, neutron beam, neutron scattering, materials testing.*

1. Introduction

Realizing the importance of nuclear technology and its potential applications in various fields, the Pakistan Atomic Energy Commission (PAEC) was formed in 1956. Since its establishment, PAEC has made considerable efforts for the socio-economic development of the country through the peaceful use of nuclear technology. In 1965, the first nuclear research reactor and in 1972 the first nuclear power plant was established in Islamabad and Karachi, respectively. The research reactor was installed at the Pakistan Institute of Nuclear Science and Technology (PINSTECH) and named Pakistan Research Reactor-1 (PARR-1). Later another reactor PARR-2 was installed at PINSTECH. PARR-1 is a 10 MW multipurpose research reactor whereas PARR-2 is a 30 kW low-power miniature neutron source reactor (MNSR).

PARR-1 is a swimming pool MTR fuel-type reactor that became critical in 1966. In 1990, it was upgraded from 5 MW to 10 MW and its fuel was also converted from high-enriched uranium (HEU) to low-enriched uranium (LEU). PARR-1 is a medium flux facility with a maximum neutron flux of $1.5x10^{14}$ $ncm^{-2}s^{-1}$ at the core. It has six radials and one tangential through neutron beam tubes. Three out of six neutron beam tubes are dedicated to neutron scattering instruments. The other two neutron beam tubes have neutron activation analysis (NAA) and neutron radiography (NR) facilities whereas one beam tube could not be utilized due to space limitations.

2. Neutron Beam Applications

Neutron beams are produced either in constant wavelength sources which are research reactors (RRs) or in pulsed sources such as spallation sources. Neutron beams have numerous applications in the fields of materials science, natural sciences, life sciences, earth sciences, industry and medicine such as isotope production and boron neutron capture therapy (BCNT). Applications of neutron beams in these areas are effectively utilized in research and development, which constructively contribute to the improvement of quality of life, societal well-being and consequently, in the overall socio-economic development of the country.

2.1. Industrial Applications

2.1.1. Neutron Transmutation Doping

The neutron transmutation, which is initiated by interaction with neutrons, changes the nucleus to another or multiple nuclides through a nuclear reaction. The resulting unstable nucleus state undergoes some processes to become stable, which has different properties as compared to the original one [1]. Doping is purposely introducing impurity atoms into the material to obtain the desired properties. Therefore, neutron transmutation doping (NTD) is a process to create impurities in the intrinsic or extrinsic semiconductor by neutron irradiation to improve its electrical properties. There are several candidate materials for NTD such as Si, Ge, GaAs, GaN, GaP, InP, InSe and HgCdTe [1]. However, commercial scale NTD is only viable with Si due to several reasons: high quality single crystal ingots are available in the market, extremely uniform and ease of doping and great demand for doped Si in power devices and sensors [2, 3].

In NTD, the n-type semiconductor is produced by the conversion of ^{30}Si to ^{31}P by the following reaction [4]:

$$Si\text{-}30(n,\gamma)Si\text{-}31 \longrightarrow P\text{-}31$$

The total cost to develop an NTD facility can reach half a million dollars, which includes the installation of irradiation rigs, handling equipment and QC instruments [2]. Si ingots are brittle therefore, careful handling before and after irradiation is required. In RRs, having NTD facility storage area of at least one ton of Si ingot should be available and in order to clean the irradiated ingot, an ultrasonic bath connected to radioactive waste water tanks is necessary [4].

Thermal neutrons are required for NTD as defects produced by fast neutron irradiation impair the electrical properties, especially at high resistivity values [1]. Therefore, thermal neutron density along the central axis is crucial and typically thermal neutron flux ~ 5×10^{12} - 10^{13} $cm^{-2}s^{-1}$ to obtain resistivity

*Corresponding author:khurram@pinstech.org.pk

of 200 Ω.cm along with a high Cd ratio is desirable. For instance, Si ingot irradiated for 17 hours at 10^{13} $cm^{-2}s^{-1}$ thermal neutron flux can produce 50 Ω.cm resistivity and large facilities with a flux $>10^{14}$ $cm^{-2}s^{-1}$ can irradiate 10-15 tons of Si ingots [1]. The time required for irradiation depends on several factors such as the choice of an appropriate irradiation position, neutronic and thermohydraulic calculations, design of the irradiation rig, calibration measurements to obtain the desired parameters and trained manpower [2].

In the early 2010s, 120-150 tons of Si ingot were doped in RRs at the cost of US $ 70-100 /kg. It is estimated if annual production of hybrid vehicles reaches 50 million by 2030 this demand will rise to 2000 tons per annum [1].

2.1.2. Gemstone Coloration

The majority of natural gems are single crystals of naturally occurring minerals, but some are amorphous (some types of opal and natural glass), others are solid solutions (e.g., garnets and peridot), some are rocks (jade and lapis etc.) and some are made entirely or mostly of organic materials (such as amber, pearls, coral) [5]. The price of a gemstone depends on several factors: color, luminosity and durability. The value of the gemstone can be enhanced with coloration. Different techniques such as neutron irradiation, ion beam and laser beam are used for gemstone coloration. Localized heating is required for inducing color in gems; however, laser beams do not produce localized heating, whereas low-mass ions produce localized heating but result in poor color [6]. Blue coloring in the colorless gemstones topaz and diamond is possible with neutron irradiation and coloration can enhance 30 times the natural value of these gemstones [1]. Gemstones are irradiated with fast neutrons therefore; the container can be shielded with B or Cd. The aluminum container holding up to 2 kg of gemstones is used for irradiation and placed in either the grid position or in the beam tube [1]. Fluence of 10^{17} - 10^{18} fast neutrons is desirable which is equivalent to an irradiation time 50-100 hours in 2MW RR. The temperature of gemstones typically varies between 100-150 °C during irradiation but damage can occur in topaz as discoloration if the temperature soars above 300 °C [2]. In some cases, 70% stones can be transported after 2 months of cooling time but β-emission should be checked with a multi-channel analyzer before releasing gemstones to ensure the dose is not more than the acceptable value of 2 nCi/g [1].

The gemstone market is volatile and difficult to access due to unpredictable trends in the fashion industry. The behavior of gemstone dealers is also erratic, primarily due to the long radiological cooling time of gemstones, which puts their investment at stake. Gemstone's market worth was US$ 14.34 billion in 2021 [7]. Last year, Pakistan exported US$ 13.73 million, which is only 0.09% of the global share [8]. Pakistan can increase its share in the gemstone market as we can exploit the natural reserves of topaz, which are found mostly in Gilgit-Baltistan [9].

2.1.3. Residual Stress Measurement

Residual stress is one of the primary factors influencing the mechanical characteristics of a material, such as strength, plasticity and surface integrity. For instance, tensile stress conditions can severely reduce component life, while compressive stress conditions can improve material performance under fatigue [10]. Therefore, the distribution of residual stress can predict the service life of the materials. Residual stress is produced in materials during fabrication processes such as plastic deformation and heat treatment. Residual stress can impair the mechanical properties and the service life of the material is reduced. Therefore, knowledge of the residual stress is extremely important to improve the thermo-mechanical processing of metals and alloys, thereby increasing the service life of a material. This technique is also utilized in the inspection of the reliability of weld and post-weld heat treatment in the high-tech industries such as the nuclear industry.

At the microscopic scale, residual stress changes the distance between atomic planes which is reflected in the shift of Bragg reflection. This change in the lattice plane spacing "*d*" can be accurately measured with neutron diffraction (ND) and X-ray diffraction (XRD). In Fig. 1A, variation of peak (020) as a result of change in *d* spacing of different brass samples can be easily seen. B2 was annealed at 200 °C, B3 at 300 °C and so on. The corresponding residual stress is presented as a function of annealing temperature in Fig. 1B.

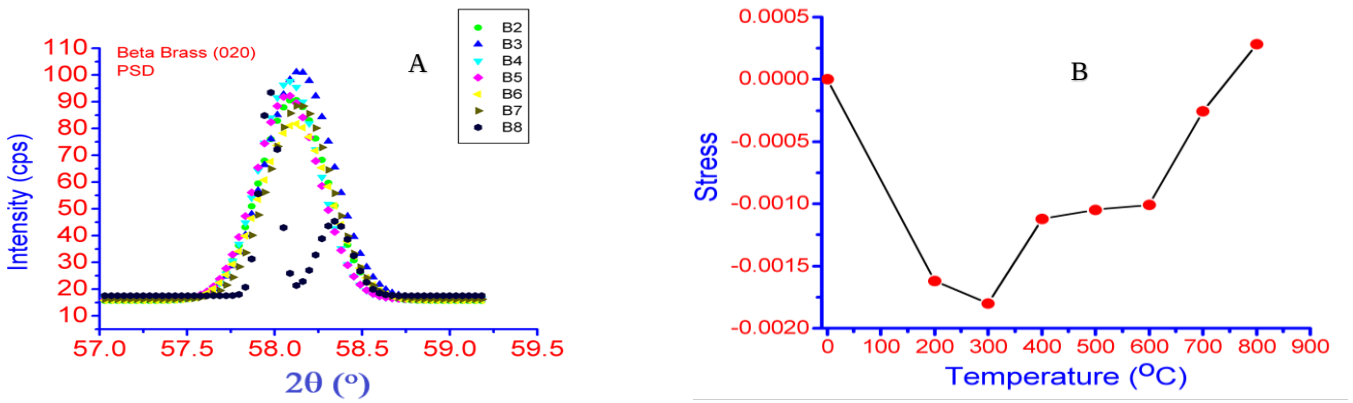


Fig. 1: (A) Change in the 2θ position of Bragg peaks in brass sample measured at different temperatures, (B) calculated residual stress as a function of temperature.

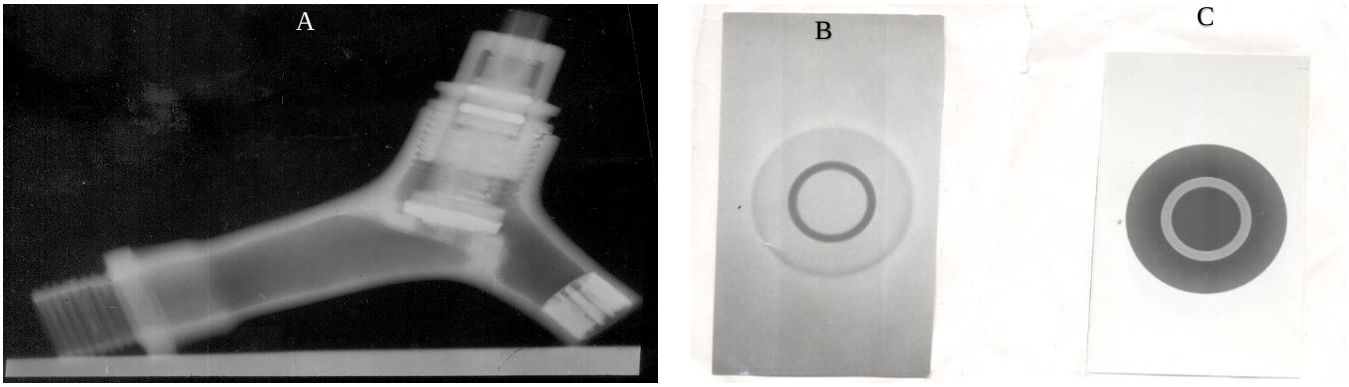


Fig. 2: Neutron radiography image of (A) water tap, (B) O-ring in grooved aluminum, (C) X-ray radiograph of O-ring in grooved aluminum.

The ND technique has the advantage of large penetration depth due to charge neutrality and small absorption of neutrons in most of the elements compared with XRD. Therefore, 3-D mapping of residual stress in the bulk of a component using a non-destructive technique is possible only with ND [11, 12].

2.1.4. Neutron Radiography

Neutron radiography (NR) is one of the most important non-destructive techniques that has applications in the fields of materials science and engineering, cultural heritage, geology, paleontology, etc. [13]. Neutrons' ability to see light elements, especially those surrounded by heavy elements makes NR superior to other techniques such as X-ray, ɤ-ray and ultrasonic testing. A variety of samples such as airplane turbine blades, oil in the engine, detonating fuse, plant root, irradiated fuel, metal fluid, boiling of high-pressure liquid, etc., can be tested with NR.

The NR setup is installed at beam tube 6 at PARR-1. Images of the water tap and O-ring are shown in Fig. 2. In Fig. 2A, the details inside the tap can be easily seen including metallic and polymeric components. Similarly, in Fig. 2B, the O-ring can be seen sitting in groves of aluminum, however, the X-ray image of the same object's O-ring is not visible (Fig. 1C).

2.2. Medical Applications

2.2.1. Radioisotope Production

A number of radioisotopes can be produced at PARR-1 on demand by medical centers that are spread across the country. Some of these radioisotopes are used to make freeze-dried kits at PINSTECH. Technetium generators, PAKGEN 99mTc, are regularly supplied to cancer hospitals all over the country. Last year revenue of more than 300 PKR million was generated from the production of radioisotopes and radiopharmaceuticals.

2.2.1. Boron Neutron Capture Therapy

^{10}B absorbs a neutron and emits α-particle by the following reaction:

$$n + {}^{10}B \longrightarrow {}^{4}He + {}^{6}Li$$

α-particle is highly ionizing, having a range in human tissue nearly equal to the diameter of a cell. Boron neutron capture therapy (BNCT) is to inject a tumor with a borated compound and irradiate it with either thermal or epithermal neutrons [2]. Surface or shallow tumors can be irradiated with thermal neutrons whereas deep tumors require epithermal neutrons. Therefore, the success of the technique is to produce better boron compounds for drug delivery. Other elements such as Gd-based compounds can also be used in this kind of therapy [2]. The reactors with powers ranging from 100 kW to 1 MW are specially adapted for BNCT [1]. Even a low-power, fast reactor of 5 kW can produce epithermal neutrons of the desired intensity [2]. Neutrons having energy 0.5 eV to 10 keV and flux 10^8-10^9 $ncm^{-2}s^{-1}$ are needed for BNCT. A filtered collimated beam free from fast neutrons and gamma rays, which can damage healthy tissues, is required for deep-seated tumors such as glioblastoma multiform. BNCT also demands the development of medical treatment protocols, which require approval from the government health authority.

2.3. Forensic Applications

2.3.1. Neutron Activation Analysis

Neutron activation analysis (NAA) is the simplest and wide application of RRs. NAA can be performed almost in any reactor, even with a power of only a few kW, which can generate a neutron flux of 10^{11} $cm^{-2}s^{-1}$. It requires post-irradiation facilities and HPGe detector with a Compton suppression system. NAA lab can analyze 10 samples per day assuming two counting facilities. At large facilities, 4000 samples per year can be performed but only 10% are mostly utilized in their annual capacity [1].

NAA has non-destructive capability, which simultaneously provides quantitative analysis of multiple elements. This technique is well suited for trace element analysis and typical detection limits of some elements are in the sub-ppm range [14]. NAA is utilized in almost every area of science and technology including archeology, studies of air-polluting, aerosol particulates, sediment studies, sulfur in coal and sugar, boron in steel, uranium, thorium and potassium exploration in salt mines, cadmium in cigarettes, determination of toxic and essential trace elements in food [15]. On the basis of performance and expertise, the NAA

laboratory in PINSTECH has been awarded Regional Resource Unit status for the South Asian Region by the IAEA.

2.3.2. Prompt Gamma Neutron Activation Analysis

Prompt gamma neutron activation analysis (PGNAA) setup is installed at beam-tube 2 at PARR-1. The advantages of PGNAA as compared to NAA are that the resulting product is stable and emits no intense gamma radiation. Further, isotopes with very short half-life or small isotopic abundance can easily be measured.

PGNAA can also perform a variety of analyses such as analysis of 316-L stainless steel, sediment studies, Cd in cigarettes and B analysis in steel [15]. Sulfur in coal from the Baluchistan coal mine has been studied with PGNAA at PINSTECH.

2.4. R&D applications

2.4.1. Neutron Scattering

Compared to other techniques, neutron scattering provides a distinct advantage for studying a wide range of materials. Thermal neutrons have a wavelength that is ideal for diffraction investigations or elastic scattering. Conversely, because the energies of phonons and neutrons are similar, lattice dynamics are analysed using inelastic scattering. Selective absorption and the absence of charge on neutrons allow for deep penetration into the materials to study their bulk properties [16]. Due to their distinct scattering lengths, neutrons can even distinguish between hydrogen and deuterium. Determining low- and high-density ice, as well as creating the extraordinarily intricate temperature and pressure phase diagram of water (H_2O-D_2O), would not have been possible without neutron scattering [17]. These advantages of neutrons also make them suitable for in-situ, operando and non-ambient measurements [18]. The magnetic moment of neutrons is another tool used to study magnetic structures of the magnetic materials. Some of these advantages of neutron scattering techniques are utilized at PARR-1.

2.4.1.1. Triple-axis diffractometer

The triple-axis diffractometer TKSN-400 displayed in Fig. 3, used for inelastic neutron scattering experiments, was installed in the early 1970s. This instrument has been extensively used for elastic scattering as well as inelastic scattering experiments. Instrumental and experimental details are provided elsewhere [11]. It has a bent Cu (220) single crystal as a monochromator and its wavelength is selectable. This diffractometer was later upgraded with IAEA technical and financial support. The up-gradation work included designing and developing of multicounter assembly consisting of 8 He-3 detectors with inter detector separation of 12° which covers the 2θ range of 5-110°, instrument control and data acquisition system. This instrument played a vital role in enhancing neutron scattering activities at PARR-1.

2.4.1.2. Double-axis diffractometer

A high-resolution double-axis neutron powder diffractometer

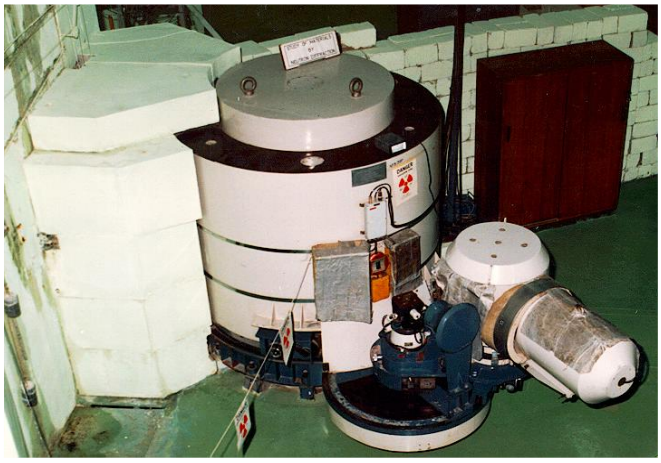


Fig. 3: Triple-axis diffractometer at beamtube 3, PARR-1.

(NPD) for elastic scattering experiments is shown in Fig. 4. NPD was installed after the upgradation of PARR-1 from 5 to 10 MW. It has a Cu (220) single crystal as a monochromator which provides neutrons of 1.27Å wavelength. 30’:20’:10’ collimation produces $1x10^5$ $n.cm^{-2}.s^{-1}$ thermal neutron flux at the sample position. It has two types of detectors, a single BF_3 counter can cover the 2θ range of 5-125° and a position sensitive detector (PSD) has active area of 2.5cmx60cm with a 2θ range of 40° are installed on the NPD.

Fig. 4: PARR-1 double-axis diffractometer installed at beamtube 4.

This diffractometer has been used to study crystal structure by determining the accurate atomic positions of light elements as well as the ordering of elements in the unit cell and magnetic structure. The properties of the materials depend on their crystal structure. Therefore, accurate crystal structure determination is pivotal to both comprehend the properties and create novel materials with enhanced properties.

2.4.1.3. Small Angle Neutron Scattering

A small angle neutron scattering (SANS) instrument shown in Fig. 5 is installed at beamtube 5. Generally, a cold neutron source produces low-energy neutrons for SANS experiments. But SANS at PINSTECH is unique in the sense

Fig. 5: Small angle neutron scattering instrument installed at beam tube 5, PARR-1.

that it utilizes thermal neutrons instead of cold neutrons. In the early 1990s, the idea was proposed that thermal neutrons could also be used for SANS experiments [19]. The SANS instrument has two monolithic channel crystals, which decouple the resolution from the divergence of the incident beam. This decoupling improves angular resolution by two orders of magnitude and a divergent beam can generate sufficient neutron intensity for experiments. In our case, the flux at the sample position is ~$5x10^4$ $n.s^{-1}.cm^{-2}$. This instrument has a monochromator of perfect bent Si (111) crystal which gives a wavelength of 2.1 Å. It has a fully asymmetrically bent perfect crystal Si (111) as an analyzer having Q-resolution = 10^{-4} - 10^{-3} $Å^{-1}$ and Q-range = $2x10^{-4}$ - $2x10^{-2}$ $Å^{-1}$ and ~ 1 mm resolution 1-D PSD. The maximum sample size, which can be measured is $5x25mm^2$. SANS is used for studying aging and fatigue effects, precipitates, cracks, voids and other inhomogeneities in materials.

2.4.2. Materials Testing

Materials' testing is also one of the important neutron beam applications. Materials testing can be applied to nuclear fuels and nuclear reactors' structural components. The testing can be performed inside the core, at the periphery or at an external location using neutron beam guides.

2.4.2.1 Structural Components Testing

Structural components of nuclear reactors are usually irradiated with neutrons, especially fast neutrons to study their aging behavior and predict their service life under the neutron fluence. Fast neutrons produce defects in the materials, which not only impair their mechanical properties but also their corrosion resistance. Typically, neutron fluence of 10^{17} cm^{-2} is required for testing electronic components and organic gaskets whereas more than 10^{21} cm^{-2} is needed for testing metallic components [2].

2.4.2.2 Nuclear Fuel Testing

Nuclear fuel testing is performed with thermal neutrons and generally, two types of testing are carried out in RRs. The first type of testing is the aging of nuclear fuel in which fuel burn up is measured at steady-state power conditions. The second type of testing is carried out under power transient conditions to investigate fuel and cladding behavior. A specialized device is needed for this type of testing in order to control changes in the neutron flux on the fuel sample. The minimum amount of thermal neutron flux 10^{13} $cm^{-2}s^{-1}$ is typically required at the location of the specimen [2].

3. Conclusion

PINSTECH is playing its part along with other contributions in the field of neutron beam applications to achieve sustainable socio-economic development in Pakistan. Large-scale radioisotope production not only reduces import costs but also has the potential to generate significant income from exporting them to other nations. NTD and gemstone color can both produce a sizeable sum of foreign exchange. On the other side, residual stress measurement analysis can save huge costs by extending the service life of the materials, whereas R&D generally improves the overall socio-economic condition. There is a lot of potential in the field of neutron beam applications and more neutron beam applications should be explored.

Acknowledgment

Technical and financial support from the IAEA for the upgradation of neutron scattering instruments is acknowledged. The authors are thankful to Dr. Usman Khurshid for providing neutron radiography images.

References

[1] "Commercial Products and Services of Research Reactors", IAEA-TECDOC-1715, 2013.

[2] "Applications of Research Reactors", IAEA Nuclear Energy Series No. NP-T-5.3, 2014.

[3] J.V. Logan, E.B. Frantz, L.K. Casias, M.P. Short, C.P. Morathd, and P.T. Webster, "Potential for neutron and proton transmutation doping of GaN and Ga_2O_3", Mater. Adv., Vol. 1, pp. 45-53, 2020.

[4] "Neutron Transmutation Doping of Silicon at Research Reactors", IAEA-TECDOC-1681, 2012.

[5] P. Voudouris, C. Mavrogonatos, I. Graham, G. Giuliani, A. Tarantola, V. Melfos, S. Karampelas, A. aterinopoulos, and A. Magganas, "Gemstones of Greece: Geology and Crystallizing Environments", Minerals, Vol. 9, pp. 461 1-29, 2019.

[6] S. Intarasiri, D. Bootkul, L.D. Yu, T. Kamwanna, S. Singkarat, T. Vilaithong, "Gemological modification of local natural gemstones by ion beams", Surface and Coatings Technology, Vol. 203, pp. 2788-2792, 2009.

[7] *Colored gemstones market outlook (2023 to 2033)* [online]. Available: https://www.futuremarketinsights.com/reports/colored-gemstones-market

[8] Naeem-uz-Zafar, *Annual Analytical Report on External Trade Statistics of Pakistan FY 2020-21* [online]. Available: https://www.pbs.gov.pk/sites/default/files/external_trade/annual_analytical_report_on_external_trade_statistics_of_pakistan_2020-21.pdf, page 8

[9] *Topaz from Pakistan* [online]. Available: https://www.mindat.org/locentries.php?m=3996&p=14323

[10] J. Guo, H. Fu, Bo. Pan, R. Kang, "Recent progress of residual stress measurement methods: A review", Chinese Journal of Aeronautics, Vol. 34, pp. 54-78, 2021

[11] "Development and applications of the technique of residual stress measurement using neutron beams", IAEA Technical Reports Series No. 477, 2014.

[12] "Measurement of residual stress using neutrons", IAEA-TECDOC-1457, 2003.

[13] W. Pornroongruengchok, S. Wonglee, S. Kotayee, and P. Thongjerm, "Upgrading Neutron Imaging Facility at TRR-1/M1 Reactor", J. Phys.: Conference Series. Vol. 2605, pp. 012002 1-6, 2023.

[14] POLLARD, A.M., HERON, C., Archaeological Chemistry. Cambridge, Royal Society of Chemistry (1996).

[15] "Use of Research Reactors for Neutron Activation Analysis", IAEA-TECDOC-1215, 1998.

[16] K. Shahzad, "An overview of neutron scattering techniques at PARR-1, Neutron scattering with low and medium flux neutron sources-Annexures", IAEA-TECDOC-1961, 2021.

[17] WENK, H. R., "Application of Neutron Scattering in Earth Sciences", The Journal of The Minerals, Metals & Materials Society, Vol.64, pp. 127-137, 2012.

[18] HEWAT, A. Neutron Diffraction and the Structure of New Materials, Encyclopaedia of Materials: Science and Technology, Pergamon, 2002.

[19] "Consultant's meeting on nuclear techniques in the development of advanced ceramic technologies", IAEA, Vienna, 1991.

The Nucleus ***61***, *No. 1 (2024) 22-30*

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Calculation of Solar Trajectory in the Sky and Solar Analemma as Observed from the Earth

Ruben Avila and S. Shoaib Raza*

Department of Thermo-fluids, Faculty of Engineering, National Autonomous University of Mexico (UNAM), Mexico D.F., C.P. 04510, MEXICO

ABSTRACT

In a previous companion paper "On the elliptical orbit of the Earth and the position of the Sun in the sky: an engineering approach," published in The NUCLEUS, we presented various computational methodologies for the position/trajectory of the Sun in the sky of an observer at Earth [1]. In this paper, the methodology for calculation of solar analemma (as observed from the earth surface) has been presented, along with an elaboration of the "Equation of Time," as called in literature [2,3]. The computational methodologies presented in the earlier paper included: 1) an analytical approach; 2) a numerical algorithm; and 3) a Solar Position Algorithm commonly abbreviated as PSA from the Spanish name of its developer Plataforma Solar de Almería [4]. In the numerical approach, Earth's momentum equation written in a polar coordinate system (r, θ) *was numerically solved. It was also demonstrated that if the Earth's momentum equation was transformed to eliminate the time dependence, it could be solved analytically. In this paper, a Cartesian coordinate system is used to calculate the coordinates of the pole star (Polaris) and its declination angle. The position vector of an observer that rotates with the Earth is calculated using a new Cartesian coordinate system, whose origin is located at the center of the Earth. The solar elevation angle and the solar azimuth angle are obtained by performing a set of rotations of this new coordinate system. Towards the end, the Equation of Time (EOT) is explained and used for calculating the solar analemma.*

Keywords: *Sun trajectory; Analemma; Solar declination angle, Solar azimuth angle; Equation of Time*

1. Introduction

The variation in the position of Sun in the sky over an observer is a natural phenomenon that has intrigued humankind since forever. The position of Sun has been correlated with the occurrence of natural phenomena (volcanic activity, storm cycles, earthquakes, etc.) [5]. The motion of the Sun has also been considered as a measure of time, or as a phenomenon that governs the agricultural yields/cycles and outbreak of pandemics [6]. Due to the increasing price of petroleum, engineers need to efficiently extract energy from renewable sources, such as winds and the Sun. If civil engineers efficiently use the solar energy, they may design reliable intelligent buildings and sustainable environments. In the near future, the task of the engineers will attain more importance; however, sometimes they do not have enough understanding of the mathematical techniques that the astronomers use to calculate the Sun's position in the sky. In this paper an attempt has been made to bridge the gap.

In some recently published papers, simplified techniques have been presented to calculate the Sun's position in the sky, however, it was pointed out that there was a need to develop and report new mathematical algorithms, suitable for amateur astronomers, students and practitioners in the field of solar energy, see for instance [7]. A simple parametric model, that describes the basic principles of the visible Sun path on the celestial sphere, has been presented in an earlier paper [8]. A review of the Sun's position algorithms has been published by Blanco et. al [9].

The Sun position algorithms are sophisticated schemes, which compute the position of the Sun in ecliptic, celestial and horizontal coordinates. Very recently, a review of the Sun position algorithms has been presented in [10]. On the internet it is also possible to find and to execute computer codes to calculate the position of the Sun in the sky, see for instance [11]. The purpose of this paper and the companion paper [1] is to present a self-contained material suitable for energy engineers to determine the Sun's position in the sky.

In section 2, methodology for the solar elevation angle, and the solar azimuth angle measured from north are presented. The Equation of Time (EOT) with its explanation is presented in section 3. Section 4 shows the calculation of analemma obtained using EOT [12]. It may be mentioned that an analemma is a diagram showing the position of the Sun in the sky as seen from a fixed location on Earth at the same mean solar time [2], and very interestingly resembles to the shape of figure 8 (eight) [3]. The results and discussion are presented in section 5.

2. Calculation of the Solar Elevation Angle and the Solar Azimuth Angle

Before presenting the calculation of solar elevation and azimuthal angles, it may be recalled from the basic geometry that the rotations of the Earth are: (i) about its axis that points towards the North star, and (ii) around the Sun in an elliptical trajectory. A fixed Cartesian coordinate system (o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$) could be defined, whose origin is located at the center of the Earth (see Fig. 1). Its plane $\hat{x}_1$- $\hat{x}_2$ is on the Earth's equatorial plane, and (a) the position vector of the Sun moves on the plane $\hat{x}_1$- $\hat{x}_3$, (b) its $\hat{x}_3$ axis points towards the star Polaris. The orientation of its $\hat{x}_1$, axis is defined together with the initial value (at $t^* = 0$) of the rotation angle ρ. In this paper, we have assumed that at $t^* = 0$, $\rho = 0$ radians. We can further define two vectors, the vector $\hat{x}_{obs}(t^*)$, which is the position vector of an observer that is located at a certain fixed latitude d on the Earth's surface, and the vector $\hat{x}^*_{Sun}(t^*)$, which is the Sun's position vector. Notice that the vector $\hat{x}_{obs}(t^*)$ rotates at the same angular velocity as the Earth. In the model, it is assumed that the Earth's rotation angle ρ is $0 \leq \rho \leq 2\pi$, where 2π radian corresponds to one day (24 hours or 86400 seconds), and the

*Corresponding author: ssraza@msn.com

dimensionless radius of the Earth is equal to one [1]. The increment of the rotation angle Δρ (which corresponds to the time step Δt = 60 s of the numerical solution) is calculated as:

$$\Delta\rho = \frac{2\pi \text{ x } 60}{24\,x\,3600} = 0.00436 \text{ radians} \quad (1)$$

The dimensionless three components of the rotating vector $\hat{x}_{obs}(t^*)$ referred to the fixed Cartesian coordinate system (o, $\hat{x}$) are given as:

$$\hat{x}^*_{1\,obs}(t^*) = \cos\delta\cos\rho(t^*)\,,\ \hat{x}^*_{2\,obs}(t^*) = \cos\delta\sin\rho(t^*)\,,$$
$$\hat{x}^*_{3\,obs}(t^*) = \sin\delta \quad (2a)$$

and the dimensionless components of the Sun's position vector $\hat{x}^*_{Sun}(t^*)$, which oscillates from $\beta(t^*) = -23.45^o$ to $\beta(t^*) = 23.45^o$ on the plane $\hat{x}_1$- $\hat{x}_3$, are the following:

$$\hat{x}^*_{1sun}(t^*) = \cos\beta(t^*)\,,\hat{x}^*_{2\,sun}(t^*) = 0, \hat{x}^*_{3\,sun}(t^*) = \sin\beta(t^*) \quad (2b)$$

2.1 *Solar elevation angle*

Solar elevation angle is the value that describes how high the Sun is. In order to calculate the solar elevation angle α_e, we transform (at each time step Δt^*) the dimensionless components of the vector $\hat{x}^*_{obs}(t^*)$, see Eq. (2a), into a new Cartesian coordinate system (o, $\hat{x}'_1$, $\hat{x}'_2$, $\hat{x}'_3$) which has the following characteristics (see Fig. 1): (i) its origin is located at the center of the Earth, (ii) its axis $\hat{x}'_1$ always points towards the Sun (that moves on the plane $\hat{x}_1$ - $\hat{x}_3$), (iii) its axis $\hat{x}'_2$ always coincides with the axis $\hat{x}_2$ of the fixed Cartesian coordinate system (o, $\hat{x}$), (iv) it oscillates (from $\beta(t^*) = -23.45^o$ to $\beta(t^*) = 23.45^o$) about its axis $\hat{x}'_2$. The dimensionless components of the vector $\hat{x}'^*_{obs}(t^*)$ are obtained by using the transformation rule explained below.

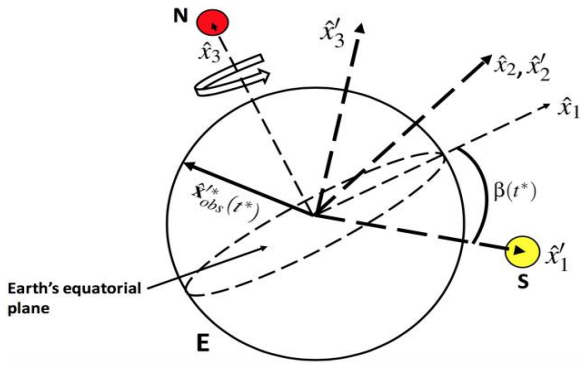


Fig. 1 Cartesian coordinate system (o, $\hat{x}'_1$, $\hat{x}'_2$, $\hat{x}'_3$), whose origin is located at the center of the Earth. Its axis $\hat{x}'_1$ always points towards the Sun. Its axis $\hat{x}'_2$ always coincides with the axis $\hat{x}_2$ of the fixed Cartesian coordinate system o, $\hat{x}$. It oscillates, from $\beta(t^*) = -23.45^o$ to $\beta(t^*) = +23.45^o$, about its axis $\hat{x}'_2$.

$$\begin{bmatrix} \hat{x}'^*_{1\,obs}(t^*) \\ \hat{x}'^*_{2\,obs}(t^*) \\ \hat{x}'^*_{3\,obs}(t^*) \end{bmatrix} =$$

$$\begin{bmatrix} \cos\beta(t^*) & 0 & \cos(90^o - \beta(t^*)) \\ 0 & 1 & 0 \\ \cos(90^\circ + \beta(t^*)) & 0 & \cos\beta(t^*) \end{bmatrix} \begin{bmatrix} \hat{x}^*_{1\,obs}(t^*) \\ \hat{x}^*_{2\,obs}(t^*) \\ \hat{x}^*_{3\,obs}(t^*) \end{bmatrix} \quad (3)$$

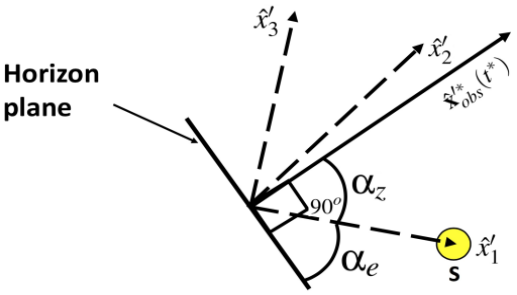


Fig. 2 The solar zenith angle α_z and the solar elevation angle α_e.

In vector form and using Eq. (2a), the Eq. (3) can be written as:

$$\hat{x}'^*_{obs}(t^*) = \big(\cos\beta(t^*)\cos\delta\cos\rho(t^*) + \cos(90^\circ - \beta(t^*))\sin\delta\big)\hat{\imath}'_1 + \cos\delta\sin\rho(t^*)\hat{\imath}'_2$$
$$+\big(\cos(90^o + \beta(t^*))\cos\delta\cos\rho(t^*) + \cos\beta(t^*)\sin\delta\big)\hat{\imath}'_3 \quad (4)$$

The dot product between the unit position vector of the observer $\hat{x}'^*_{obs}(t^*)$ and the unit vector $\hat{\imath}'_1$ that points towards the Sun, is obtained as (see Fig. 2):

$$\hat{x}'^*_{obs}(t^*)\cdot\hat{\imath}'_1 = |\hat{x}'^*_{obs}(t^*)||\hat{\imath}'_1|\cos\alpha_z$$
$$= \cos\beta(t^*)\cos\delta\cos\rho(t^*) + \sin\beta(t^*)\sin\delta \quad (5)$$

where, the trigonometric identity $\cos(90^\circ - \beta(t^*)) = \sin\beta(t^*)$ has been used. From Eq. (5) the solar zenith angle α_z is obtained as:

$$\alpha_z = \cos^{-1}[\cos\beta(t^*)\cos\delta\cos\rho(t^*) + \sin\beta(t^*)\sin\delta] \quad (6)$$

The solar elevation angle α_e is obtained as:

$$\alpha_e = 90^\circ - \alpha_z \quad (7)$$

2.2 *The solar azimuth angle measured from north*

The solar azimuth angle is the value that describes in which direction of the Sun is from north of the observer's horizon plane. The solar azimuth angle is obtained by performing two rotations of the Cartesian coordinate system (o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$). After the two rotations (represented by the transformation matrices A and B) have been carried out (Fig.3) the unit vector $\bar{\bar{\imath}}_3$ of the new Cartesian coordinate

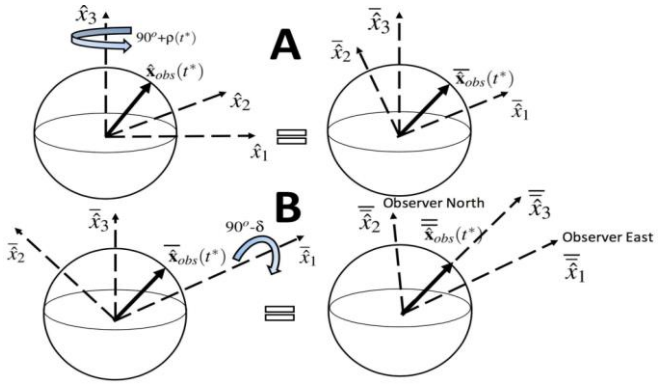


Fig. 3 Transformation matrices A and B from the Cartesian coordinate system o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$ to the Cartesian coordinate system o, $\bar{\bar{\hat{x}}}_1$ $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$.

system (o, $\bar{\bar{\hat{x}}}_1$ $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$) coincides with the position vector of the observer $\hat{x}^*_{obs}(t^*)$ see Eq. (2a), while the units vectors $\bar{\bar{\hat{\imath}}}_1$ and $\bar{\bar{\hat{\imath}}}_2$ are directed to the east and north directions, respectively, of the observer. Note that the plane defined by the two unit vectors $\bar{\bar{\hat{\imath}}}_1$ and $\bar{\bar{\hat{\imath}}}_2$ is (in a natural way) the horizon plane of the observer.

By performing this procedure, the Sun's position vector $\hat{x}^*_{Sun}(t^*)$, see Eq. (2b), that is defined in the coordinate system (o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$), is transformed into the coordinate system (o, $\bar{\bar{\hat{x}}}_1$ $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$). The angle between the unit vector, $\bar{\bar{\hat{\imath}}}_2$, that is directed to north of the observer, and the projection of the vector $\bar{\bar{\hat{x}}}^*_{Sun}(t^*)$, on the plane $\bar{\bar{\hat{\imath}}}_1$ - $\bar{\bar{\hat{\imath}}}_2$ is the solar azimuth angle, α_a.

2.2.1 The transformation matrices A and B

The first rotation is about the axis $\hat{x}_3$ of the coordinate system (o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$). If the rotation angle is 90° + ρ(t*), the first transformation matrix A is as follows:

$$A = \begin{bmatrix} -\sin\rho(t^*) & \cos\rho(t^*) & 0 \\ -\cos\rho(t^*) & -\sin\rho(t^*) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

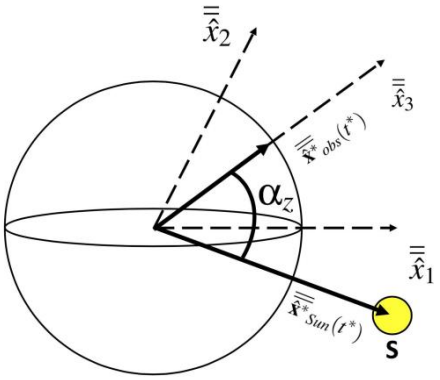


Fig. 4 The observer position vector $\bar{\bar{\hat{x}}}^*_{obs}(t^*)$, the Sun position vector $\bar{\bar{\hat{x}}}^*_{Sun}(t^*)$ and the solar zenith angle α_z in the Cartesian coordinate system o, $\bar{\bar{\hat{x}}}_1$, $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$.

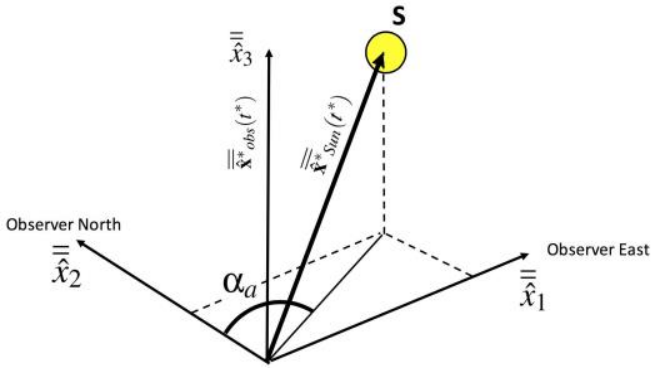


Fig. 5 The observer position vector $\bar{\bar{\hat{x}}}^*_{obs}(t^*)$, the Sun position vector $\bar{\bar{\hat{x}}}^*_{sun}(t^*)$ and the solar azimuth angle measured from north α_a in the Cartesian coordinate system o, $\bar{\bar{\hat{x}}}_1$, $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$. The observer horizon plane is defined by the plane $\bar{\bar{\hat{x}}}_1$ - $\bar{\bar{\hat{x}}}_2$, where $\bar{\bar{\hat{x}}}_1$ is the observer's East while $\bar{\bar{\hat{x}}}_2$ is the observer North.

After the first rotation, the Cartesian coordinate system (o, $\overline{\hat{x}_1}$, $\overline{\hat{x}_2}$, $\overline{\hat{x}_3}$) is obtained (top panel of Fig. 3). The second rotation is about the axis $\overline{\hat{x}_1}$ of the new coordinate system. If the rotation angle is 90° - δ, the second transformation matrix B is defined as follows:

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin\delta & \cos\delta \\ 0 & -\cos\delta & \sin\delta \end{bmatrix} \quad (9)$$

After the second rotation, the Cartesian coordinate system (o, $\bar{\bar{\hat{x}}}_1$, $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$) is obtained, see bottom panel of Fig. 3. Now, the axis $\bar{\bar{\hat{x}}}_3$ coincides with the position vector of the observer $\hat{x}^*_{obs}(t^*)$, and the axes $\bar{\bar{\hat{x}}}_1$ and $\bar{\bar{\hat{x}}}_2$ define the observer's horizon plane. The axis $\bar{\bar{\hat{x}}}_1$ is now directed towards the east, while the axis $\bar{\bar{\hat{x}}}_2$ is directed to the north of the horizon plane.

The global transformation matrix from the coordinate system (o, $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$) to the coordinate system (o, $\bar{\bar{\hat{x}}}_1$, $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$) is defined as R = BA, which is written as:

$$R = BA = \begin{bmatrix} -\sin\rho(t^*) & \cos\rho(t^*) & 0 \\ -\sin\delta\cos\rho(t^*) & -\sin\delta\sin\rho(t^*) & \cos\delta \\ \cos\delta\cos\rho(t^*) & \cos\delta\sin\rho(t^*) & \sin\delta \end{bmatrix} \quad (10)$$

The next step is to transform the Sun's position vector $\hat{x}^*_{Sun}(t^*)$, see Eq. (2b), into the new coordinate system (o, $\bar{\bar{\hat{x}}}_1$, $\bar{\bar{\hat{x}}}_2$, $\bar{\bar{\hat{x}}}_3$), hence the following transformation is carried out:

$$\bar{\bar{\hat{x}}}_{Sun}(t^*) = \begin{bmatrix} -\sin\rho(t^*) & \cos\rho(t^*) & 0 \\ -\sin\delta\cos\rho(t^*) & -\sin\delta\sin\rho(t^*) & \cos\delta \\ \cos\delta\cos\rho(t^*) & \cos\delta\sin\rho(t^*) & \sin\delta \end{bmatrix} \begin{bmatrix} \cos\beta(t^*) \\ 0 \\ \sin\beta(t^*) \end{bmatrix} \quad (11)$$

or,

$$\bar{\bar{\hat{x}}}_{Sun}(t^*) = \begin{bmatrix} -\sin\rho(t^*)\cos\beta(t^*) \\ -\sin\delta\cos\rho(t^*)\cos\beta(t^*) + \cos\delta\sin\beta(t^*) \\ \cos\delta\cos\rho(t^*)\cos\beta(t^*) + \sin\delta\sin\beta(t^*) \end{bmatrix} \quad (12)$$

From the three components of the vector $\bar{\bar{\hat{x}}}^*_{Sun}(t^*)$, the solar zenith angle, α_z (hence the elevation angle α_e), see Fig. 4, and the solar azimuth angle measured from north of the observer's horizon (see Fig. 5) are obtained.

The third direction cosine of the vector $\bar{\bar{\hat{x}}}^*_{Sun}(t^*)$ see Eq. (12), represents the cosine of the angle between the position vector of the observer and the Sun's position vector, that is, the cosine of the zenith angle α_z, which has been previously evaluated, see Eq. (6). The solar azimuth angle measured from north of the observer's horizon is obtained by taking the tangent of the angle measured from the axis $\bar{\bar{\hat{x}}}_2$ (north direction) to the projection of the vector $\bar{\bar{\hat{x}}}^*_{Sun}(t^*)$ on the plane $\bar{\bar{\hat{x}}}_1$ - $\bar{\bar{\hat{x}}}_2$ that is:

$$\tan\alpha_a = \frac{\bar{\bar{\hat{x}}}^*_{1\,Sun}(t^*)}{\bar{\bar{\hat{x}}}^*_{2\,Sun}(t^*)} \quad (13)$$

hence, the solar azimuth angle measured from north (i.e. α_a), see Eq. (12), is given as

$$\alpha_a = \tan^{-1}\left\{\frac{-\sin\rho(t^*)\cos\beta(t^*)}{\cos\delta\sin\beta(t^*) - \sin\delta\cos\rho(t^*)\cos\beta(t^*)}\right\} \quad (14)$$

In Fig. 6, a two dimensional map is showing the solar azimuth angle α_a at the abscissa and the solar elevation angle, α_e at the ordinate, for four days (March 20, June 21, September 19 and December 21) of the year 2013, corresponding to a latitude $\delta = 40.73^o$. The declination angle (β) is used to calculate the angles α_a and α_e. The declination angle (β) at the beginning of the day and at the end of the day (see the values of Declin. start and Declin. end), as well as the sunrise and the sunset regions, are displayed in each panel of Figure. Notice that for the latitude $\delta = 40.73^o$ (and for the locations on the Earth's northern hemisphere with latitude greater than 23.45^o) the Sun is always south of the place (see the values of Declin. start and Declin. end in Fig. 6), hence the solar azimuth angle α_a measured from north of the observer's horizon plane at noon is always $\alpha_a = 180^o$ (south direction of the observer's horizon plane). In Fig. 6, it is observed that the sunrise region is located in the interval $0^o < \alpha_a < 180^o$ (north-east-south direction of the observer's horizon plane), while the sunset region is in the interval between $180^o < \alpha_a < 360^o$ (south-west-north direction of the observer's horizon plane). The elevation angle α_e ($\alpha_e \geq 0^o$) is measured from the location at which α_a is evaluated. Then in Fig. 6, if α_a is lower than 90^o or if α_a is greater than 270^o, the Sun is north of the observer's horizon plane and α_e is measured from north of this plane. On the other hand, if α_a is in the interval $90^o < \alpha_a < 270^o$ the Sun is south of the observer's horizon plane, then α_e is measured from south of this plane. However, for the locations on the Earth's north hemisphere with latitude $\delta < 23.45^o$ (not shown here), in some dates of the year, the Sun is north of the place, hence the physical interpretation of the angles α_e and α_a is different. That is, when the Sun is north of the place, the solar azimuth angle measured from north of the observer's horizon plane is in the interval $-90^o < \alpha_a < 90^o$ degrees. Where, the interval from 90^o to 0^o corresponds to the region from sunrise to noon, while the interval from 0^o to -90^o corresponds to the region from noon to sunset.

3. Equation of Time

The results of the Sun's position in the sky of an observer, presented in the previous sections, are not fully correlated with the observations of one person that attempts to see the position of the Sun at the same clock time every day along the year. This observer will notice that the Sun is not at the same position in the sky. The reason is that the clock time is based on a standard measure that considers a fictitious Earth that travels around the Sun in a circular trajectory with constant tangential velocity (hence with constant angular velocity). Additionally, the standard measure of time (clock time) takes into account that the Sun is always located at the Earth's equator. As it was previously shown, the trajectory of the true Earth is elliptical, and its tangential velocity v_t as well as the

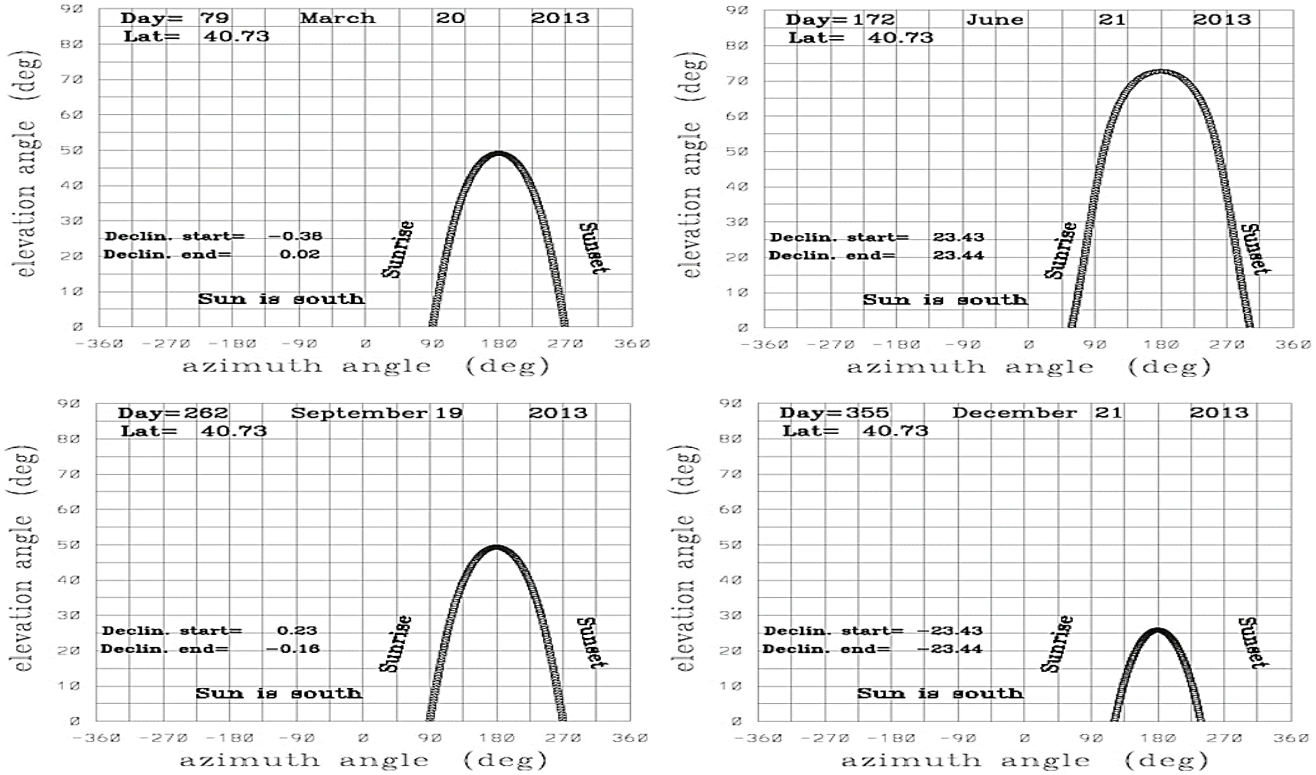


Fig. 6 Map of the solar azimuth angle is measured from north of the observer's horizon plane α_a (see Eq. 14) and the solar elevation angle α_e is calculated from Eq. (7). The observer is located at a latitude $\delta = 40.73^o$. Four days of the year 2013 are shown. The number of the day along the year, the sunrise, and the sunset regions are displayed. The declination angle β at the beginning of the day in degrees (Declin. start) and the declination angle β at the end of the day in degrees (Declin. end) are also displayed. The legend Sun is south, means that the declination angle β is lower than the latitude of the place $\delta = 40.73^o$.

Sun's position relative to the Earth's equator (the declination angle β), are both functions of time. The difference between the standard measure of time (clock time) and the Sun's position in the sky of an observer (solar time) is known as the equation of time or also called correction of time.

3.1 Standard measure of time (clock time)

In the standard measure of time model, it is assumed that a fictitious Earth with constant angular velocity Ω_{Circ} follows a circular trajectory. The angular velocity Ω_{Circ} is given by:

$$\Omega_{Circ} = 360/365 = 0.98630136 \text{ [Degrees/day]} \quad (15a)$$

or,

$$\Omega_{Circ} = 360/365 \times 1/24 \times 1/60 = 0.0006849315 \text{ [Degrees/min]} \quad (15b)$$

where, [Degrees] is the unit of angle measured along the circular trajectory.

In the standard measure of time, it is also assumed that the fictitious Earth's angular velocity ω_{Circ} about its rotation axis, which is normal to the ecliptic plane, is 360 degrees per day, or:

$$\omega_{Circ} = 360/24 \times 1/60 = 0.25 \text{ [degrees/min]} \quad (16)$$

where, [degrees] is the unit of angle of the fictitious Earth's rotation about its axis. In the clock time model, it is assumed that the position of the Sun relative to the position of an observer is exactly the same each 24 hours (i.e. each 360 degrees or each 0.98630136 Degrees). The Sun's position relative to the observer in the circular trajectory is evaluated by considering that the fictitious Earth travels on a circle with unit radius *R*. At the center of the circle is the Sun. As shown in Fig. 7, the origin of the fixed coordinate system (O, X_1, X_2) is located at the center of the circle (at the Sun's position), while the origin of the moving and locally rotating coordinate system (o, x_1, x_2) is located at the Earth's position $X_E(t)$.

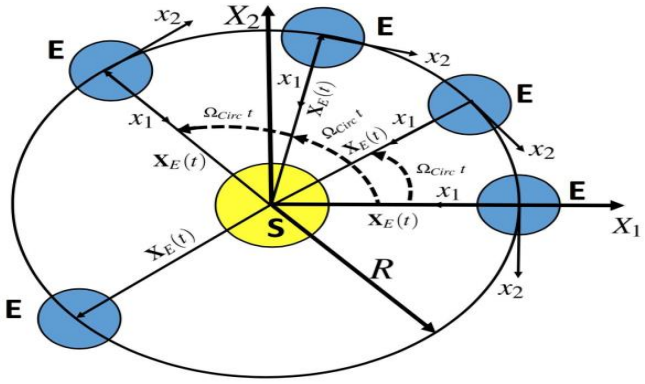


Fig. 7 Circular trajectory of the fictitious Earth: two Cartesian coordinate systems are defined. The fixed coordinate system (O, X_1, X_2) whose origin is located at the center of the circle with unit radius R (at the Sun's position), and the moving and rotating coordinate system (o, x_1, x_2), whose origin is located along the circle, i.e. at the Earth's position vector $X_E(t) = R \cos(\Omega_{Circ}\, t)\, I_1 + R \sin(\Omega_{Circ}\, t)\, I_2$. The axis x_1 always is directed to the Sun, hence it is in the same direction of the vector $X_E(t)$.

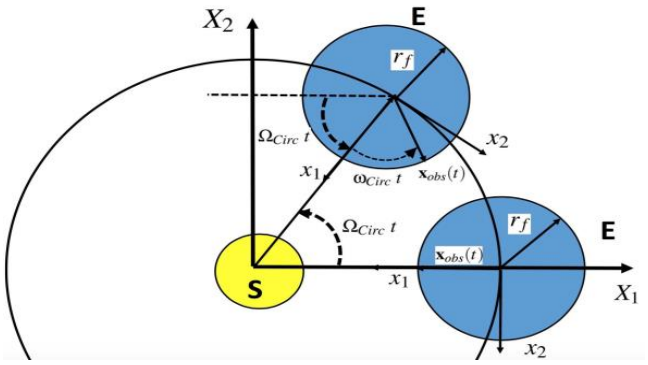


Fig. 8 Circular trajectory of the fictitious Earth: the moving coordinate system (o, x_1, x_2), rotates by an angle Ω_{Circ} t. In the moving coordinate system, the position of an observer on the fictitious Earth surface is given as $x_{obs}(t) = r_f \cos((\omega_{Circ} + \Omega_{Circ})t)\, i_1 + r_f \sin((\omega_{Circ} + \Omega_{Circ})t)\, i_2$, where r_f is the unit radius of the fictitious Earth.

In the fixed coordinate system the position vector of the Earth is defined as:

$$X_E(t) = R\cos\left(\Omega_{Circ}\, t\right) I_1 + R\sin\left(\Omega_{Circ}\, t\right) I_2, \quad (17)$$

where, I_1 and I_2 are the unit vectors of the fixed coordinate system and t is the elapsed time in minutes. In the fictitious Earth model, it is assumed that the coordinate system (o, x_1, x_2) is also rotating an angle Ω_{Circ} t, in such a way that the axis x_1 is always in the direction of the vector $X_E(t)$, see Figs. 7 and 8. In the moving and rotating coordinate system, the position of an observer on the fictitious Earth' surface is given as:

$$x_{obs}(t) = r_f \cos\left((\omega_{Circ} + \Omega_{Circ})t\right) i_1 + r_f \sin\left((\omega_{Circ} + \Omega_{Circ})t\right) i_2 \quad (18)$$

where, i_1 and i_2 are the unit vectors of the moving and rotating coordinate system and r_f is the unit radius of the fictitious Earth. In the fixed coordinate system (O, X_1, X_2), the position vector $X_{obs}(t)$ of the observer can be evaluated through the use of the two components of the position vector $x_{obs}(t)$ (see Eq. 18), which are defined on the surface of the fictitious Earth and in the coordinate system (o, x_1, x_2). The position vector $X_{obs}(t) = X_E(t) - x_{obs}(t)$ (see Fig. 8), is calculated as:

$$X_{obs}(t) = \left(R\cos(\Omega_{Circ}\, t) - r_f \cos\left((\omega_{Circ} + \Omega_{Circ})t\right)\right) I_1 + \left(R\sin(\Omega_{Circ}\, t) - r_f \sin\left((\omega_{Circ} + \Omega_{Circ})t\right)\right) I_2 \quad (19)$$

Now, in the fixed Cartesian coordinate system we have the following three vectors: $X_E(t)$, $X_{obs}(t)$ and X_{E-obs}. The last vector is the relative vector from the center of the fictitious Earth to the observer. The relationship between the three vectors (that is $X_{E-obs}(t) = X_{obs}(t) - X_E(t)$), is given as follows:

$$X_{E-obs}(t) = -r_f \cos\left((\omega_{Circ} + \Omega_{Circ})t\right) I_1 - r_f \sin\left((\omega_{Circ} + \Omega_{Circ})t\right) I_2 \quad (20)$$

3.2 First component of the equation of time: correction due to eccentricity of the Earth's orbit

It is clear from the previous discussion that the true Earth (that moves in an elliptical trajectory) has an angular velocity ($\Omega_{Ell}(t) = d\theta(t)/dt$), that is not a constant. As presented earlier, two Cartesian coordinate systems are defined to represent the elliptical path of the true Earth. It is assumed that the rotation axis of the true Earth is normal to the ecliptic plane. The origin of the fixed coordinate system (O, X_1, X_2) is located at the focus of the ellipse (at the Sun's position), while the origin of the moving coordinate system (o, x_1, x_2) is located at the center of the Earth that is moving along its elliptical orbit. In the fixed coordinate system the position vector of the true Earth is given as:

$$\mathrm{X}_E(t) = r^*(t)\cos\theta(t)\mathrm{I}_1 + r^*(t)\sin\theta(t)\mathrm{I}_2, \qquad (21)$$

where, r*(t) is the dimensionless polar coordinate. If we assume that the moving coordinate system also rotates as a function of time (i.e the angle θ(t)), it is explicitly considered that the axis x_1 is always in the direction of the vector $X_E(t)$, then the position of the observer on the surface of the true Earth is given as:

$$\mathrm{x}_{obs}(t) = r_t\cos\big(\omega_{\mathrm{Circ}}\, t + \theta(t)\big)\mathrm{i}_1 + r_t\sin\big(\omega_{\mathrm{Circ}}\, t + \theta(t)\big)\mathrm{i}_2, \qquad (22)$$

where, r_t is the unit radius of the true Earth. In the fixed coordinate system (O, X_1, X_2), the coordinates of the observer, which are defined in the moving coordinate system (see Eq. 22), are obtained as:

$$\begin{aligned}\mathrm{X}_{obs}(t) &= \mathrm{X}_E(t) - \mathrm{x}_{obs}(t)\\ &= \Big(r^*(t)\cos\theta(t) - r_t\cos\big(\omega_{\mathrm{Circ}}\, t + \theta(t)\big)\Big)\mathrm{I}_1 +\\ &\quad \Big(r^*(t)\sin\theta(t) - r_t\sin\big(\omega_{\mathrm{Circ}}\, t + \theta(t)\big)\Big)\mathrm{I}_2.\end{aligned} \qquad (23)$$

In the fixed Cartesian coordinate system we have the three vectors: $X_E(t)$, $X_{obs}(t)$ and X_{E-obs}. It is worth mentioning that the graphs, as a function of time, of the vectors defined by Eqs. (21)-(23), confirm again the fact that the observer (even if the trajectory is elliptical with variable angular velocity $\Omega_{Ell}(t)$) sees, along the year, the Sun at the same position at the same hour of the day, as it was the case of the circular trajectory with constant angular velocity Ω_{circ}. The reason of this behavior of the position vectors $X_E(t)$, $X_{obs}(t)$ and X_{E-obs}, in both cases (the fictitious Earth and the true Earth) is because the moving coordinate system (o, x_1, x_2), whose axis x_1 is always in the same direction of the vector $X_E(t)$, is rotating with an angular velocity (either Ω_{circ} or Ω_{Ell}), which is the same as the angular velocity of the Earth's trajectory (either circular or elliptical).

The first part of the Equation of Time is calculated by using the following equation:

$$\Delta t_{\mathrm{first}} = \frac{\Omega_{\mathrm{Circ}}\, t - \theta(t)}{\omega_{\mathrm{Circ}}} \equiv [\text{ minutes }] \qquad (24)$$

In Fig. 9, the left panel shows the first component of the Equation of Time (correction due to the eccentricity of the Earth's orbit). The angle θ(t) in Eq. (24) is obtained by using the polar angle θ(t), which is the independent variable, and the PSA algorithm (for the year 2013), in which instead of using the polar angle θ(t), the ecliptic longitude $\hat{\theta}$ is used. In the PSA algorithm, instead of using the angle spanned by the fictitious Earth, $\Omega_{\mathrm{Circ}}\, t$, the mean longitude L is used, while in the analytical approach, the time t in the expression $\Omega_{\mathrm{Circ}}\, t$, is obtained by the analytical solution. In the left panel of Fig. 9, it is observed that when the Earth is close to the Sun (January, February), the first component of the Equation of time is negative, i.e. the angular velocity of the true Earth ($\Omega_{Ell}(t) = d\theta(t)/dt$) is higher than the constant angular velocity of the standard fictitious Earth (Ω_{Circ}), as it was shown in the results of our imaginary scenery. However, when the Earth is far from the Sun (July, August) the angular velocity of the true Earth is lower than that of the fictitious Earth, then the first correction to the Equation of time is positive.

3.3 Second component of the equation of time: correction due to the Earth's rotation axis tilt

The correction of time due to the tilt of the Earth's rotation axis relative to the ecliptic plane, is based on the variation, with respect to the time of the declination angle β(t). In order to understand the contribution of the tilt angle on the equation of time, we propose a model based on two Earths: (1) the

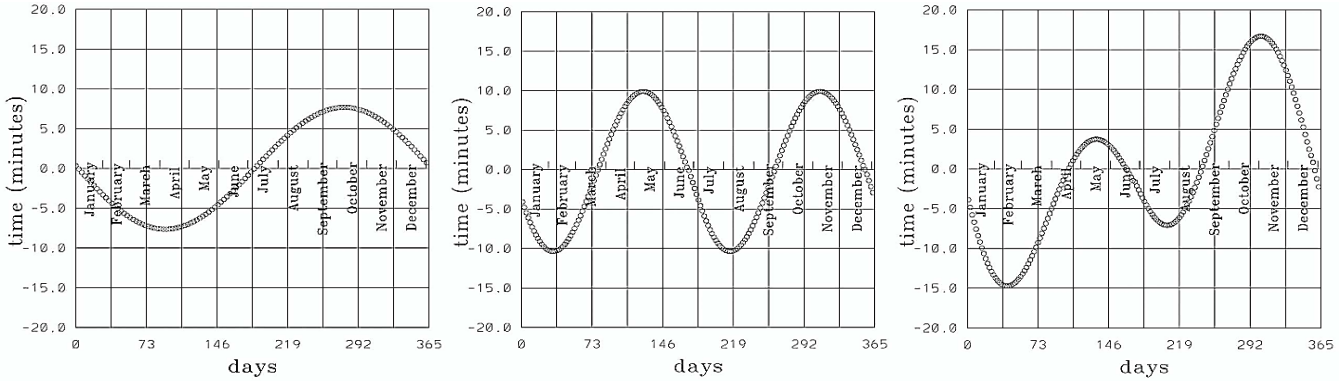


Fig. 9. The Equation of Time. Left panel: correction due to the eccentricity of the Earth's orbit, see Eq. (24); PSA algorithm results for 2013, and analytical results (see text). Middle panel: correction due to the tilt (obliquity) of the Earth's rotation axis from the normal to the ecliptic plane. Right panel: the Equation of Time as the sum of the two corrections shown in the left and middle panels, see Eq. (35).

standard fictitious Earth, and (2) the true Earth. In this model, it is assumed that as the Earth rotates about its rotation axis, Sun draws a curved line on the surface of each Earth. On one hand, in the fictitious Earth, it is assumed that, for a certain initial condition (noon of one selected day, that is when $\rho = 0$ radians), the latitude angle $\hat{\beta}$ (which is the value of the declination angle $\beta(t)$ at noon of the selected day) remains constant as the Earth rotates about its rotation axis, from noon of the selected day to noon of the next day (that is from $\rho = 0$ radians to $\rho = 2\pi$ radians). Along this fictitious day, it is apparent that the Sun generates on the surface of the Earth, a closed circular path that is parallel to the Earth's equatorial plane. The radius $\hat{r}$ (that is normal to the fictitious Earth's rotation axis) of this closed circular path is given as:

$$\hat{r} = r_f \cos\hat{\beta}, \tag{25}$$

where, r_f is the unit radius of the fictitious Earth. Then, after 24 hours, from noon of the selected day, and maintaining constant the latitude angle $\hat{\beta}$, the Sun has traveled on the surface of the fictitious Earth a total circular distance $\hat{d}_{f\ ict}$ given by

$$\hat{d}_{\text{fict}} = 2\pi\hat{r} = 2\pi r_f \cos\hat{\beta} \tag{26}$$

On the other hand, the closed curved path of the Sun on the surface of the true Earth is governed, not only by the longitude angle $\rho(t)$, but also by the declination angle $\beta(t)$. For the true Earth model, let us assume again as initial condition, that at noon (that is when $\rho = 0$ radians) of the day that was selected for the fictitious Earth, the Sun is at the declination angle $\beta(t)$, that is equal to the latitude angle $\hat{\beta}$ of the fictitious Earth (which remains constant until the next noon). Then, between two times $t_1$and t_2 (where $\Delta t = t_2 - t_1 = 1$ minute, corresponds to the time increment that the Sun's position (on the surface of the true Earth), has changed, not only due to the rotation of the Earth (given by the angle $\rho(t)$), but also due to the variation of the declination angle $\beta(t)$. To calculate the Sun's curved path on the surface of the true Earth between the two points P_1 (at t_1) and P_2 (at t_2), the spanned angle σ_{12} (the central angle) between these two points is evaluated either as:

$$\sigma_{12} = \cos^{-1}\left(\sin\beta_1 \sin\beta_2 + \cos\beta_1 \cos\beta_2 \cos(\rho_2 - \rho_1)\right), \tag{27}$$

or as:

$$\sigma_{12} = 2\sin^{-1}\left(\frac{C}{2}\right) \tag{28}$$

where, the variable C is given as:

$$C = \sqrt{(\Delta X)^2 + (\Delta Y)^2 + (\Delta Z)^2} \tag{29}$$

and

$$\Delta X = \cos\beta_2 \cos\rho_2 - \cos\beta_1 \cos\rho_1,$$
$$\Delta Y = \cos\beta_2 \sin\rho_2 - \cos\beta_1 \sin\rho_1\,, \Delta Z = \sin\beta_2 - \sin\beta_1 \tag{30}$$

The sub-indexes 1 and 2 mean that the angles $\beta(t)$ and $\rho(t)$ are evaluated at the times t_1 and t_2, respectively. The longitude angle spanned by the Earth's rotation after 1 minute is given by $\Delta\rho = \rho_2 - \rho_1 = 0.00436$ radians $\equiv 0.25$ degrees.

The distance d^{12} (arc length) traveled by the Sun on the surface of the true Earth, between the two points P_1 and P_2, is obtained as:

$$d^{12} = r_t \sigma_{12}, \tag{31}$$

where, r_t is the unit radius of the true Earth. Then, after 24 hours (or after $r=2\pi$ radians), the Sun has generated on the surface of the true Earth an helical (spiral) closed curved path whose total length $\hat{d}_{\text{true}}$ is given by the sum of each of the small distances d^{12}, traveled by the Sun, between each pair of points P_1 and P_2. As $\Delta t = 1$ minute, one day is constituted by 1440 time increments, hence along one day we have np=1440 pair of points P_1 and P_2. Hence, the total helical (spiral) distance traveled by the Sun on the surface of the true Earth (along 1440 minutes), is given as:

$$\hat{d}_{\text{true}} = \sum_{i=1}^{np} d_i^{12} \tag{32}$$

In the previous analysis, it has been considered that the radii r_f (see Eq. 26) and r_t (see Eq. 31) of both Earths is unitary, then both of the distances $\hat{d}_{\text{fict}}$ and $\hat{d}_{\text{true}}$ are expressed in radians, hence the difference $\hat{d}_{\text{diff}} = \hat{d}_{\text{fict}} - \hat{d}_{\text{true}}$ must be converted to minutes by using the following relationship:

$$\hat{d}_{\text{diff}} = \left(\hat{d}_{\text{fict}} - \hat{d}_{\text{true}}\right)[\text{ radians }]$$
$$= \left(\frac{24}{2\pi}\right)(60)\left(\hat{d}_{\text{fict}} - \hat{d}_{\text{true}}\right) \equiv [\text{ minutes }] \tag{33}$$

Consequently, the time correction due to the Earth's rotation axis tilt for one day that begins at $\rho(t)=0$ radian, and at $\hat{\beta} = \beta(t)$ (for the fictitious Earth) and at $\beta(t)$ (for the true Earth), and finishes (after 24 hours or 1440 minutes) at $\rho(t) = 2\pi$ radians and at $\hat{\beta}$ (for the fictitious Earth) and at the corresponding declination angle $\beta(t)$ for the true Earth, is given by the following equation:

$$\Delta t_{\text{second}} = \pi^2 \hat{d}_{\text{diff}} \equiv [\text{ minutes }] \tag{34}$$

where, the scaling factor π^2 is introduced to reproduce the second component of the Equation of Time published in the literature. Middle panel of Fig. 9 shows the second component of the Equation of time (see Eq. 34) as a function of time (days of the year). It is observed that the second correction of time is zero for four days: the two equinoxes and the two solstices. In these particular days, the Sun generates on the surface of the true Earth, circular paths (not spirals) similar to the circular paths generated by the Sun on the surface of the fictitious Earth. In the middle panel of Fig. 9, it is also observed that when the Sun travels from south to north, from the March equinox, to the June solstice (that is from March 20 to June 21), the daily spiral distance traveled by the Sun, $\hat{d}_{\text{true}}$ on the surface of the true Earth, is smaller than the daily distance, $\hat{d}_{\text{fict}}$ traveled by the Sun on the surface of the fictitious Earth, then the Sun on the true Earth, is ahead from the Sun on the fictitious Earth, hence the second correction of time Δt_{second} is positive. The same situation appears when the Sun travels from north to south, from the September equinox to the December solstice (from Sept. 2019 to Dec.2021).

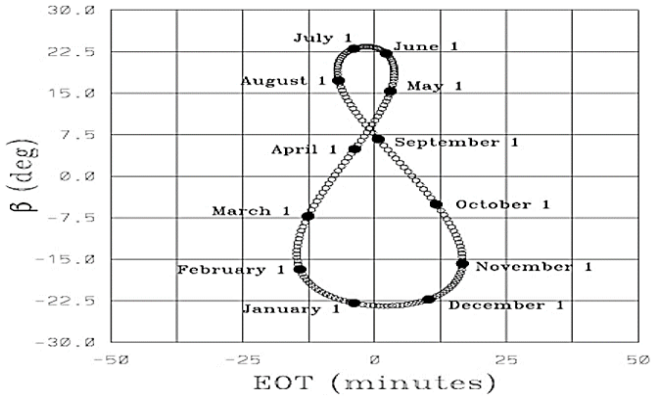


Fig. 10 Analemma for the year 2013. The Equation of Time, see Eq. (35) (as the abscissa) and the declination angle β(t), as the ordinate.

However, when the Sun is traveling from north to south, from the June solstice to the September equinox (that is from June 21 to September 19), the daily spiral distance traveled by the Sun, $\tilde{d}_{\text{true}}$, on the surface of the true Earth, is greater than the daily distance $\hat{d}_{\text{fict}}$, traveled by the Sun on the surface of the fictitious Earth, then the Sun on the true Earth, is delayed with respect the Sun on the fictitious Earth, hence the second correction of time is negative. The same situation appears when the Sun travels from south to north, from the December solstice to the March equinox (from December 21 to March 20). The sum of the times Δt_{first} , see Eq. (24) and Δt_{second}, see Eq. (34), is the so called Equation (correction) of Time given by:

$$EOT = \Delta t_{\text{first}} + \Delta t_{\text{second}} \equiv [\text{ minutes }] \tag{35}$$

This Equation of Time (EOT) is plotted in Fig. 9 (right most panel) as function of time (days along the year).

4. Solar Analemma

An analemma is a diagram showing the position of the Sun in the sky as seen from a fixed location on Earth at the same mean solar time. As the solar position varies over the course of a year, a line joining the solar position, for the same date and time of every month of the year resembles a figure like 8 (eight) [2,3]. Figure 10 shows the graph of the solution of the Equation of Time, see Eq. (35) (as the abscissa) and the declination angle, β(t) (as the ordinate) at noon (ρ = 0 radians) of each day along the year 2013 to yield the corresponding analemma. Note that the analemma of Fig. 10, does not depend on the latitude at which an observer is located on the Earth's surface. Fig. 11, shows the analemma calculated for an observer located at the latitude δ=40.73º for the year 2013. Left panel shows the analemma at 10:00 A.M., middle panel shows the analemma at noon, and the right panel shows the analemma at 4:00 P.M. It is important to point out that the solar azimuth angle shown as the abscissa in the panels of Fig. 11, is the modified solar azimuth angle $\hat{\alpha}_{\text{a}}$. This angle is obtained by correcting Eq. (14) with the Equation of Time (EOT) (see Eq. 35). That is, the abscissa of the analemma of Fig. 11, is obtained by using the following relationship:

$$\hat{\alpha}_{\text{a}} = \alpha_{\text{a}} + (0.25)\ \text{EOT} \tag{36}$$

where, the coefficient 0.25 is the conversion factor from minutes to degrees.

5. Results and Discussion

In an earlier companion paper diverse computational methodologies were presented to calculate the trajectory of the Sun in the sky of an observer located on the Earth's surface. In this paper, the location of the North Star has been successfully calculated in a Cartesian coordinate system, which is a familiar coordinate system for the engineers.

Additionally, the position vector from the Earth to the Sun, γ(t) and the declination angle, β(t) as functions of time, were obtained by using an engineering approach. Standard transformations of the involved vectors (the position vector of the observer and the Sun's position vector) have been obtained by performing simple rotations of the Cartesian coordinate system. The horizon plane of the observer, the solar zenith angle α_{z}, the solar elevation angle α_{e}, and the solar azimuth angle measured from north of the observer's horizon plane α_{a} have been obtained by using the proposed simple engineering approach. It has been pointed out that when the Sun is north of the observer, a different interpretation of the

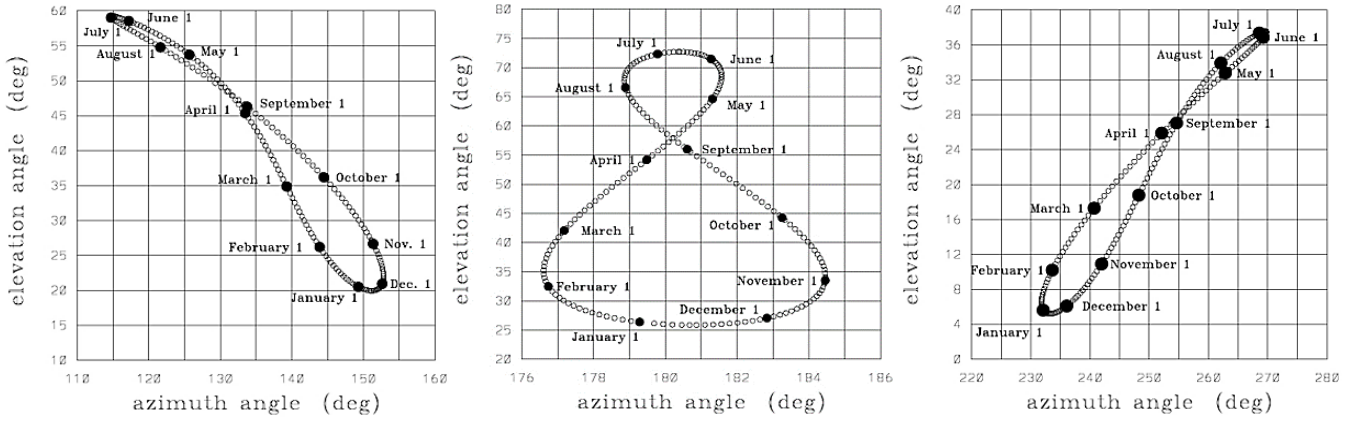


Fig. 11 Analemma for the year 2013. The observer is located at a latitude δ=40.73º. The solar azimuth angle measured from north α_{a} (degrees), see Eq. (14), as the abscissa and the solar elevation angle α_{e} (degrees), see Eq. (7), as the ordinate. Left panel: Analemma at 10:00 A.M. Middle panel: Analemma at noon. Right panel: Analemma at 4:00 P.M.

solar azimuth angle is needed, that is, when the declination angle $\beta(t)$ is greater than the latitude of the observer δ, the solar azimuth angle $\alpha_a(t)$ must be defined within the interval $-90^{o} < \alpha_a(t) < 90^{o}$

The two components of the Equation of Time have been explained in detail (from the physical point of view), and a new relationship to obtain the correction of time due to the Earth's rotation axis tilt has been presented.

For a graphic representation of the Equation of Time, see Fig. 9, and for a plot of the solar declination angle as a function of time, see Fig. 10.

The analemma for three distinct hours of the day (of the year 2013), for an observer located at the Earth's latitude $\delta = 40.73^{o}$ have been shown (see Fig. 11, corresponding to the New York city). The methodology presented here can be easily applied to other latitude locations and other calendar years.

The information included in this paper should be considered as an important source of reference for the solar energy engineers, who need to accurately know the position of the Sun throughout the year. The equations presented in this paper can be easily used by engineers instead of the sophisticated mathematical equations used by astronomers and astrophysicist.

Acknowledgment:

The authors are grateful to the anonymous reviewers who contributed in the improvement of this manuscript.

References

[1] R. Avila and S. R. Syed, "On the elliptical orbit of the Earth and position of the Sun in the sky: an engineering approach", The Nucleus, vol. 61, no. 1, pp. 10–15, 2024.

[2] https://ysjournal.com/astrophysics/analemma. [Accessed 5 August, 2020]

[3] M. Ihtirozun and K. Suraya, "Correlation of Analemma's Position Towards the Beginning of Maghrib Prayer." Al-Hilal: Journal of Islamic Astronomy, Vol. 3, No. 1, 2021.

[4] I. Reda, and A. Andreas, "Solar position algorithms for solar radiation applications." Tech. Rep. NREL/TP-560-34302, National Renewable Energy Laboratory, Golden, Colorado, USA, January 2008.

[5] M. Lockwood, M. Owens, Ed Hawkins, G.S. Jones, I. Usoskin, "Frost fairs, sunspots and the Little Ice Age," Astronomy & Geophysics, vol. 58 (2), 2017.

[6] T. Rogers, "A History of Agriculture and Prices in England." Vol. 7, Part 1. Oxford: Clarendon Press, 1902.

[7] A. Jenkins, "The Sun's position in the sky". Eur. J. Phys., 34, pp. 633–652, 2013.

[8] V. Khavrus and I. Shelevytsky, "Introduction to solar motion geometry on the basis of a simple model". Phys. Ed., vol. 45, pp. 641–653, 2010.

[9] B. Muriel, A. Padilla, L. Moratalla and L. Coira, "Computing the solar vector". Solar Energy, vol. 70, pp. 431–441, 2001.

[10] G. Prinsloo and R. Dobson, Solar Tracking. Stellenbosch: Solar Books, 2015.

[11] National Oceanic and Atmospheric Administration US Department of Commerce" NOAA Solar Calculator". http://www.esrl.noaa.gov/ gmd/grad/solcalc/.[Accessed 5 August, 2020]

[12] D. Anthony, "Earth's orbit". http: //bado-shanai.net/Astrogation/. [Accessed 5 August, 2020]

The Nucleus ***61***, *No. 1 (2024) 31-39*

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Feasible Size Ratio Prediction of Wind and PV Module for Energy Generation of Different Climatic Zones in India

Ranjana Khandare and Rubina Chaudhary*

School of Energy & Environmental Studies, Devi Ahilya University, Indore, Madhya Pradesh, India

ABSTRACT

Solar and wind are renewable energy resources that are non-exhaustible, freely available, ecofriendly and capable for providing solutions to the power problems, which are recently being faced by India and all over the world. With increasing demand of generating electricity, wind and solar energies are not available all the time, the combined implementation of renewable resources such as wind and solar can meet the increasing demand of electricity. This Paper presents study about the Photovoltaic-Wind Hybrid System (PWHS) under six climate zones of India. The study is based on the size ratio of Average Solar Irradiation Energy (ASIE) and Average Wind Speed Energy (AWSE). Annually, the ASIE varied form 5.00 KWh/m^2 to 6.50 KWh/m^2 whereas the values of AWSE varied between 0.1 m/s and 2.4 m/s. After analyzing different climatic conditions, PWHS was designed. Through mathematically modeling and learning programming techniques, different size ratios were conformed. Linear programming was applied on average annual value of photovoltaic (PV) and wind, to determine the size ratio of PV wind among different climate zones.

Keywords: *Hybrid System; Wind Speed; Photovoltaic; Solar Irradiation*

1. Introduction

The availability of solar and wind energy resources is determined by the climatic conditions at specific location. Pre-feasibility studies depend upon the climatic data such as the wind speed, solar irradiance and load demand for that particular location. In order to calculate the performance of a system, an appropriate weather data is required. The weather conditions may vary from one location to another. Because of the large variation of weather conditions in different parts, India needs to focus on hybrid power projects, which support optimizing the production and power at competitive prices as well as to reduce the variability. Also, to formulate the mandatory standards and regulations for hybrid systems which are not included in the previously announced policy (wind-solar hybrid policy on May 15, 2018). The hybridization of two or more renewable systems along with conventional power source battery storage can increase the performance of the renewable technologies as reported earlier [1]. Advantage of the hybrid energy systems is that the energy can be exploited from several sources at the same time which will raise the overall efficiency of the system as reported by Das et al. earlier [2]. In the past few years, increasing sustainable growth in the industrial sector and global population has increased the energy demand all over the world. Presently, conventional power systems mainly rely on the usage of nonrenewable resources such as oil, gas, and coal. Therefore, utilization of these fossil fuels contribute to several negative effects on the environment due to the emission of greenhouse gases as reported by Sharvini [3].

The aim of this study is to propose the Photovoltaic-Wind Hybrid Systems (PWHS) which are applicable all over the India as suggested by Luthra, et al. [4]. Meteorological data of six different locations of India which come under different climatic zones was utilized. The main idea behind this study was to find the size ratio of photovoltaic and wind system (PVWS) by using linear programming algorithms [5].

1.1. Meteorological Data

Global solar irradiance and possible wind speed data was utilized for this study which is available for the year 2022. The solar irradiation data was collected from the online website wetherspark.com and wind speed data from the world weather online.com [6]. This data of six different locations in India was used to complete this work. The data was compiled on monthly average values of the global solar irradiance (KWh/m^2) and wind speed (m/s) that was utilized for the size ratio of photovoltaic and wind under the present study [7].

1.2. Climatic zones characteristic of India

The climate classification system divides the country into six major climatic zones. These zones are: (1) hot and dry, (2) warm and humid, (3) moderate, (4) cold and cloudy, (5) cold and sunny, and (6) composite. From each of the climatic zone, a representative location was selected as reported by Bhatnagar et al. [8]. The selected locations under the present study along-with their respective climatic zones and geographical data are presented in Table 1.

Table 1: Geographical parameters for selected locations in India

S.No	Climate Zone	Name of Place	Latitude	Longitude
1	Composite	Indore (Madhya Pradesh)	22º 71'	75º 85'
2	Hot and dry	Jaipur (Rajasthan)	26º 92'	75º 77'
3	Warm and humid	Dibrugarh (Arunachal Pradesh)	28º 21'	94º 72'
4	Moderate	Bangalore (Karnataka)	12º 97'	77º 59'
5	Cold and cloudy	Shimla	31º 10'	77º 13'
6	Cold and sunny	Jammu	32º 72'	74º 85'

*Corresponding author: rubinachaudhary028@gmail.com

2. Methodology

2.1. Mathematical calculation of the output power of photovoltaic generator

A photovoltaic system converts sunlight radiation into electrical energy. The system used for this conversion is the photovoltaic cell in which silicon is the basic material used for manufacturing the photovoltaic cells [9]. An electric circuit is formed when silicon wafer is connected to the electric terminals. When light from sun reaches the photovoltaic surface, the cell generates charge carriers, resulting in the electric current that flows through the short-circuit as discussed by AL-Ezzi et al. [10]. The output of PV generator (KWh) can be calculated based on Eq. (1). This equation is used by many groups working on mathematical modeling of photovoltaic power generation as reported by Ramli et al. [11]. The power output of the PV array is expressed as:

$$P_{PV}(t) = P_R.f_R.[G_T(t)/G_S] \quad (1)$$

Where P_R is PV module rated power (KW) output, f_R is loss factor or derating factor (in %) of PV module due to dirt, shadow and temperature variation etc. Derating factor (f_R) of 80% was employed in this study. G_T is the hourly global solar radiation on PV module surface (KW/m^2), and G_S is the standard incident radiation (1000W/m^2).

2.2 Mathematical calculation of output power of wind turbine

The output power of a wind turbine at specific location depends on many factors, including the amount of available energy that increases with increasing wind speed. It follows a cubic relationship between the wind energy output and its speed. At high altitude, the speed of wind is high and consistence. There is also an impact of the air temperature at high altitude. It has been noticed that cold air will result in the higher level of energy. The wind energy is used as an input source of the wind turbine and output is the mechanical power turning the generator rotor**.** The wind turbine output power can be expressed by equation (2). This equation is used in many reports in the literature related to mathematical modeling of the wind power generation system**.**

$$P_{wind} = \frac{1}{2}C_P \rho A V^3 \quad (2)$$

Where P_{wind} is the output power of the wind turbine, ρ is the air density (approximately1.23 kg/m^3), A is the area swept by the rotor blades (m^2) perpendicular to the wind velocity that can be calculated by $A= \pi RD^2/4$. Where RD is the rotor diameter (m), V is the wind speed in (m/s), and C_P is the turbine power coefficient (dimensionless) that varies between 0.2 and 0.5.

3. Predicting the best size ratio of PV-Wind hybrid system for different climate zones

3.1 Monthly average solar radiation (KWh/m^2) of six different climate zones

The first location is the composite climate of Indore, the city of Madhya Pradesh in India. Second location deals with hot and dry climate of Jaipur, the city of Rajasthan. Third location includes the warm and humid climate of Dibrugarh, the city of Arunachal Pradesh. Fourth location is the moderate climate of Bangalore, the city of Karnataka. Fifth location is comprised of cold and cloudy climate of Shimla and sixth location contains cold and sunny climate of Jammu. All these cities belong to India. Table 2 represents the monthly average solar radiation of these designated locations measured in KWh/m^2 in 2022.

3.2. First location (Indore city with composite climate)

Table 3 represents the average wind speed of Indore, Madhya Pradesh that was recorded by the world weather online between January and December 2022. The data has also been plotted in Fig.1. Indore has composite climate, which was studied in this work. Fig. 2 represents the average annual solar irradiation of the location, which comes out to be 5.80 kWh/m^2. This data along with Table 4 show the average annual wind speed that was 2.4 m/s, and the output electric power of photovoltaic generator was 0.93 W. Similarly, the output electric power of wind generator was 0.6 W. According to the linear programming, the output Photovoltaic-Wind Hybrid System (PWHS) was 1.5 W. It has been found that the size ratio of photovoltaic and wind are 40% and 60%, respectively.

Table 2: Monthly average solar radiation measured in KWh/m^2 in 2022 in different cities of India

Months	Indore	Jaipur	Dibrugarh	Bangalore	Shimla	Jammu
Jan	4.9	4.4	4.1	6.0	6.0	3.3
Feb	5.9	5.4	4.9	6.7	6.6	4.1
Mar	6.8	6.4	5.7	7.0	6.9	5.3
Apr	7.3	7.2	6.1	6.6	6.4	6.7
May	7.5	7.6	6.4	5.7	5.6	7.1
Jun	6.3	7.3	5.9	5.0	4.8	7.8
Jul	5.1	6.1	5.6	4.9	4.8	7.0
Aug	4.9	5.8	5.4	5.1	5.2	6.4
Sep	5.7	6.1	5.0	5.1	5.4	6.1
Oct	5.7	5.6	4.8	5.2	5.3	6.1
Nov	5.0	4.6	4.4	5.4	5.4	3.9
Dec	4.6	4.1	3.9	5.6	5.5	3.2

Table 3: Monthly wind speed during 2022 with an average speed of 2.4 m/s at the city of Indore

Months	Wind speed (m/s)
Jan	2.3
Feb	2.1
Mar	2.0
Apr	2.2
May	4.3
Jun	4.0
Jul	3.5
Aug	3.2
Sep	2.4
Oct	2.1
Nov	1.5
Dec	2.2

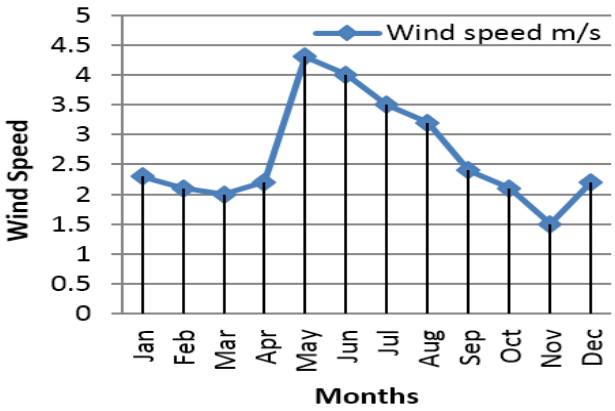


Fig. 1: Monthly average wind speed of Indore, Madhya Pradesh which was recorded between January to December 2022

Table 4: Output power of wind generator, photovoltaic, and hybrid generator at Indore for the year 2022

Months	Wind speed (m/s)	Solar irradiation (kWh/m^2)	Wind turbine on different wind speeds	Photovoltaic generator on different solar radiation	Wind turbine photovoltaic generator
Jan	2.3	4.9	2.3	0.91	3.21
Feb	2.1	5.9	1.7	1.1	2.8
Mar	2	6.8	1.5	1.2	3.2
Apr	2.2	7.3	2.0	1.3	3.3
May	4.3	7.5	15	1.4	16.4
Jun	4.0	6.3	12	1.17	13.17
Jul	3.5	5.1	8.0	0.95	8.95
Aug	3.2	4.9	6.29	0.91	7.2
Sep	2.4	5.7	2.6	1.06	3.66
Oct	2.1	5.7	1.7	1.06	2.76
Nov	1.5	5.0	0.6	0.93	1.53
Dec	2.2	4.6	2.0	0.86	2.86

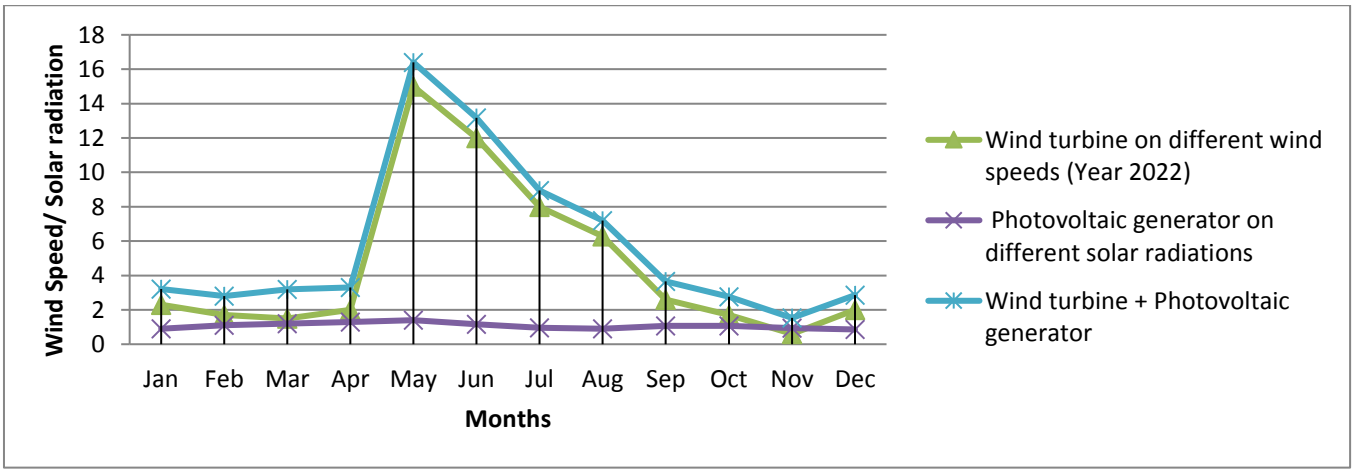


Fig. 2: Output data of wind generator, photovoltaic hybrid generator graph of Indore, Madhya Pradesh (India) as recorded by linear programming from January to December 2022

3.3. Second location (Jaipur city, hot and dry climate)

The city of Jaipur (Rajasthan) with hot and dry climate was analyzed. The average annual solar radiation recorded in 2022 was 5.80 kWh/m^2 and average annual wind speed was 2.27 m/s. Table 5 and 6 along with Fig. 3 and 4 represent the relevant data at the city of Jaipur. It is noticed that the output electric power of photovoltaic generator was monitored as 0.86 W. Similarly, the output electric power of wind generator was 0.42 W. According to the linear programming, the output

Photovoltaic-Wind Hybrid System (PWHS) was 2.15 W. It has been seen that the size ratios of photovoltaic and wind were 30% and 70%, respectively.

Table 5: Average wind speed at Jaipur & Rajasthan recorded on monthly basis in 2022

Months	Wind speed (m/s)
Jan	1.9
Feb	2.2s
Mar	2.1
Apr	2.2
May	3.4
Jun	3.4
Jul	2.4
Aug	2.7
Sep	2.5
Oct	1.8
Nov	1.3
Dec	1.4

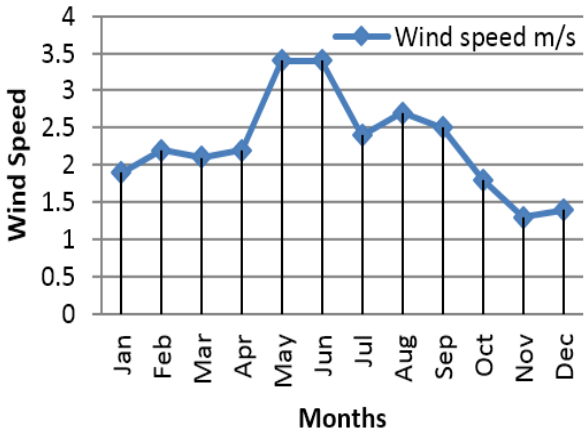


Fig. 3: Monthly average wind speed graph of Jaipur & Rajasthan which was recorded by world weather online in 2022

Table 6: Output values of wind generator, photovoltaic, hybrid generator recorded in 2022

Months	Wind speed m/s	Solar irridation kWh/m^2	Wind turbine on different wind speeds	Photovoltaic generator on different solar radiations	Wind turbine photovoltaic generator
Jan	1.9	4.4	1.3	0.748	2.04
Feb	2.2	5.4	2.0	1.01	3.01
Mar	2.1	6.4	1.7	1.19	2.89
Apr	2.2	7.2	2.0	1.34	3.34
May	3.4	7.6	7.54	1.4	9.04
Jun	3.4	7.3	7.54	1.3	8.84
Jul	2.4	6.1	2.65	1.1	3.75
Aug	2.7	5.8	3.77	1.0	4.77
Sep	2.5	6.1	3.0	1.14	4.14
Oct	1.8	5.6	1.11	1.04	2.15
Nov	1.3	4.6	0.42	0.86	1.28
Dec	1.4	4.1	0.52	0.7	1.22

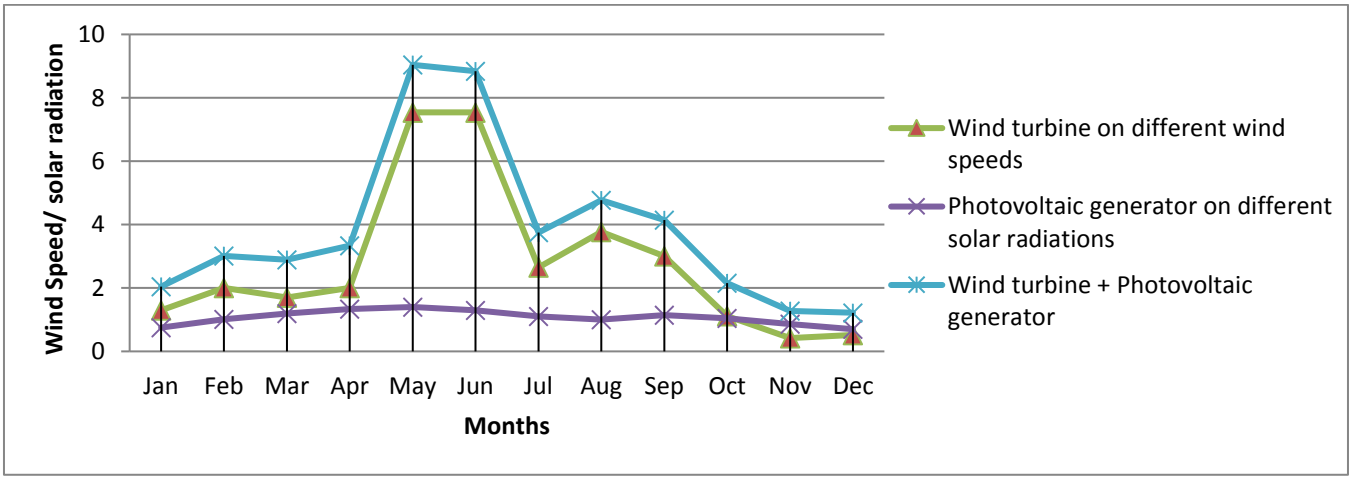


Fig. 4: Output solar radiation of wind generator, photovoltaic, hybrid generator graph recorded from January to December 2022

3.4. Third location (Dibrugarh city, warm and humid climate)

The Dibrugarh (Arunachal Pradesh) in India possessing warm and humid climate was analyzed. The average annual solar wind generator was 0.42 W. Irradiation was 5.1 kWh/m^2 and average annual wind speed was 1.77 m/s. The output electric power of photovoltaic generator was 0.89 W.

Table 7: Average wind speed of Dibrugarh (India) recorded in 2022

Months	Wind speed m/s
Jan	1.4
Feb	1.9
Mar	1.9
Apr	2.4
May	2.2
Jun	2.2
Jul	1.8
Aug	1.8
Sep	2.1
Oct	1.3
Nov	1.2
Dec	1.1

According to linear programming the output Photovoltaic-Wind Hybrid System (PWHS) was 1.31 W. It was found that the size ratio of photovoltaic and wind were 30% and 70%, respectively. Table 7 and 8 along with Fig. 5 and 6 represent the relevant data discussed in this section.

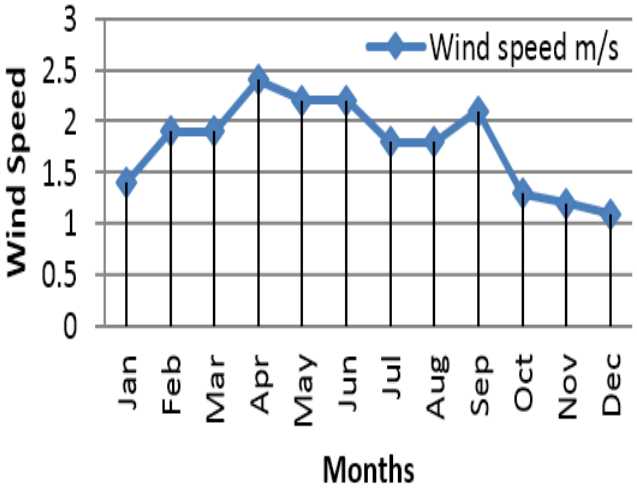


Fig. 5: Monthly average wind speed graph of Dibrugarh that was recorded by world weather online

Table 8: Output power of wind generator, photovoltaic, hybrid generator in 2022

Months	Wind speed m/s	Solar Irridation kWh/m^2	Wind turbine on different wind speeds	Photovoltaic generator on different solar radiations	Wind turbine Photovoltaic generator
Jan	1.9	4.4	1.3	0.748	2.04
Feb	2.2	5.4	2.0	1.01	3.01
Mar	2.1	6.4	1.7	1.19	2.89
Apr	2.2	7.2	2.0	1.34	3.34
May	3.4	7.6	7.54	1.4	9.04
Jun	3.4	7.3	7.54	1.3	8.84
Jul	2.4	6.1	2.65	1.1	3.75
Aug	2.7	5.8	3.77	1.0	4.77
Sep	2.5	6.1	3.0	1.14	4.14

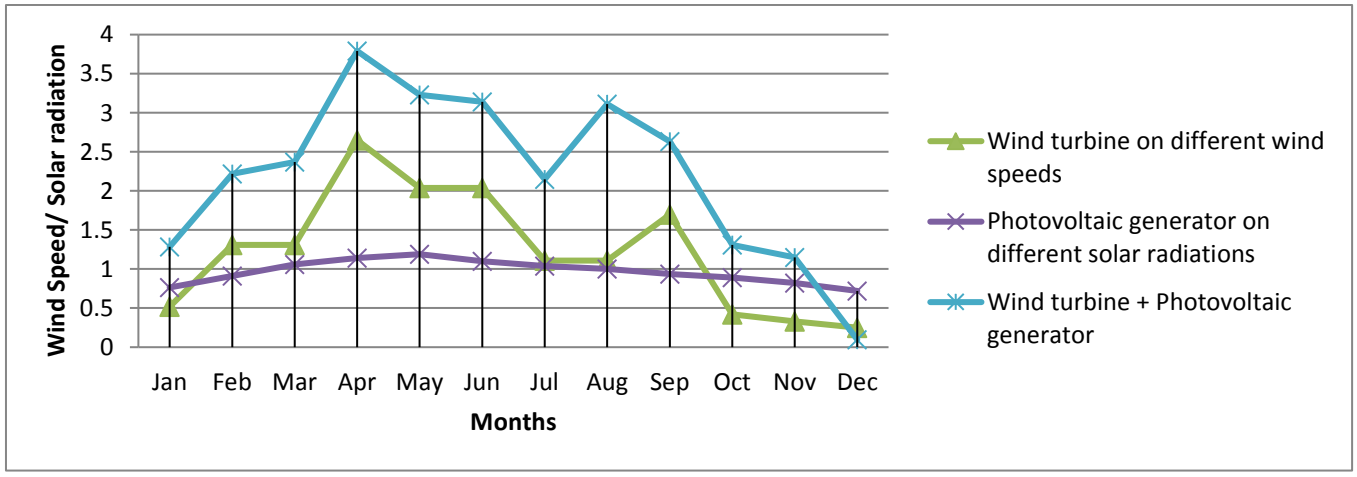


Fig. 6: Output values of wind generator, photovoltaic, hybrid generator graph of Dibrugarh (Arunachal Pradesh), recorded by linear programming between January to December 2022

3.5 Fourth location (Karnataka city, moderate climate)

Analysis of Bangalore (Karnataka) with moderate climate was presented. Table 9 and 10 along with Fig. 7 and 8 represent the relevant data collected at Karnataka. On the basis of this data, the average annual solar radiation was 5.6 KWh/m^2 and average annual wind speed was 1.8 m/s. The output electric power of photovoltaic generator was 1.30 W.

Table 9: Represents the average wind speed of Bangalore (India) recorded between January-December 2022

Months	Wind speed m/s
Jan	1.4
Feb	1.9
Mar	1.9
Apr	2.4
May	2.2
Jun	2.2
Jul	1.8
Aug	1.8
Sep	2.1
Oct	2.0
Nov	1.5
Dec	1.4

Similarly, the output electric power of wind generator was 1.33 W. According to the linear programming, the output Photovoltaic-Wind Hybrid System (PWHS) was 2.58 W. It was found that the size ratio of photovoltaic and wind are the same as 50%.

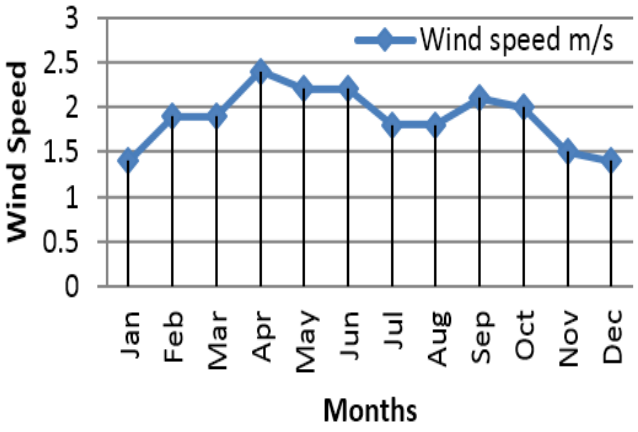


Fig. 7: Average wind speed graph of Bangalore (India) recorded by world weather online between January and December, 2022

Table 10: Output values of wind generator, photovoltaic, hybrid generator for 2022

Months	Wind speed m/s	Solar Irridation kWh/m^2	Output power of wind turbine on different wind speed	Output power of Photovoltaic generator on different solar radiation	Output power of wind turbine Photovoltaic generator
Jan	1.4	6.0	0.52	1.12	1.64
Feb	1.9	6.7	1.33	1.25	2.58
Mar	1.9	7.0	1.33	1.30	2.63
Apr	2.4	6.6	2.65	1.23	3.88
May	2.2	5.7	2.0	1.06	3.06
Jun	2.2	5.0	2.0	0.935	2.93
Jul	1.8	4.9	1.11	0.91	2.02
Aug	1.8	5.1	1.11	0.95	2.06
Sep	2.1	5.1	1.77	0.95	2.72
Oct	2.0	5.2	1.53	0.97	2.50
Nov	1.5	5.4	0.64	1.00	1.64
Dec	1.4	5.6	0.52	1.04	1.56

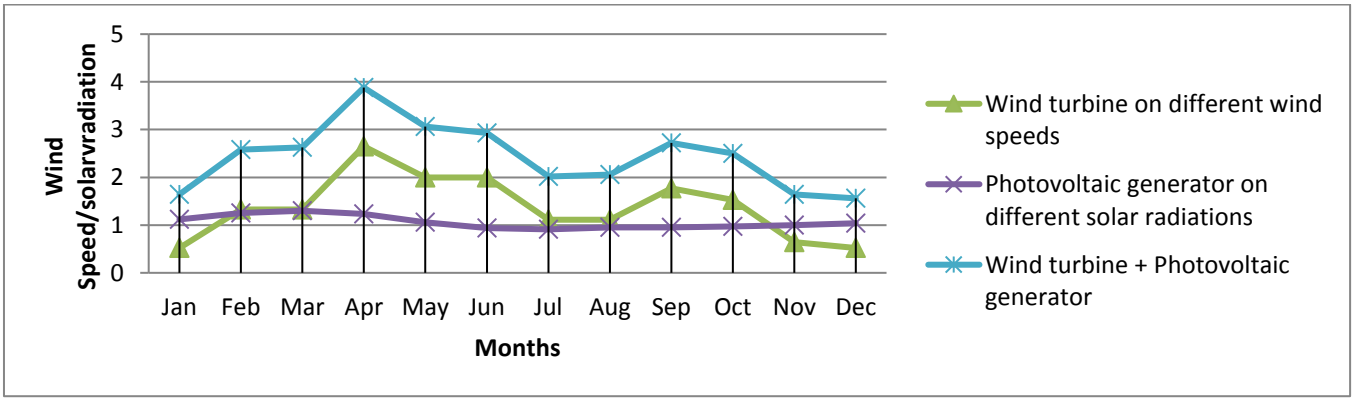


Fig. 8 Output value of wind generator, photovoltaic, hybrid generator graph of Bangalore (India) for the year 2022

3.6 Fifth location (Shimla city, cold and cloudy climate)

Shimla is having cold and cloudy climate which is analyzed. It is seen that the average annual solar radiation was 5.6 KWh/m^2 in 2022 and average annual wind speed was 0.1 m/s. The output electric power of photovoltaic generator was 1.29 W. Similarly, the output electric power of wind generator was 0.0015 W. By linear programming the output of Photovoltaic-Wind Hybrid System (PWHS) was 1.29 W and it is expressing that sizing and ratio of photovoltaic wind hybrid system in this location is not possible because the wind speed is very low. Table 11 and 12 along with Fig. 9 and 10 represent the relevant data discussed in this section.

Table 11: Aaverage wind speed of Shimla (India), which was recorded from January to December 2022

Months	Wind speed m/s
Jan	0.1
Feb	0.2
Mar	0.2
Apr	0.4
May	0.2
Jun	0.3
Jul	0.0
Aug	0.0
Sep	0.1
Oct	0.0
Nov	0.1
Dec	0.1

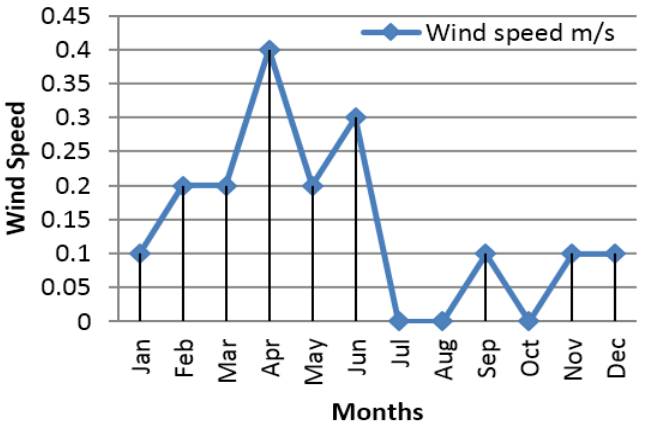


Fig. 9: Monthly average wind speed at Shimla (India) which was recorded by world weather online in 2022

Table 12: Output values of wind generator, Photovoltaic, hybrid generator

Months	Wind speed m/s	Solar Irridation kWh/m^2	Wind turbine on different wind speeds	Photovoltaic generator on different solar radiations	Wind turbine + Photovoltaic generator
Jan	0.1	6.0	.0001	1.122	1.122
Feb	0.2	6.6	.0015	1.234	1.235
Mar	0.2	6.9	.0015	1.290	1.291
Apr	0.4	6.4	.0122	1.196	1.208
May	0.2	5.6	.0015	1.047	1.048
Jun	0.3	4.8	.0051	0.897	0.902
Jul	0.0	4.8	.0000	0.897	0.897
Aug	0.0	5.2	.0000	1.028	1.028
Sep	0.1	5.4	.0001	1.009	1.009
Oct	0.0	5.3	.0000	0.991	0.991
Nov	0.1	5.4	.0001	1.009	1.009
Dec	0.1	5.5	.0001	1.028	1.028

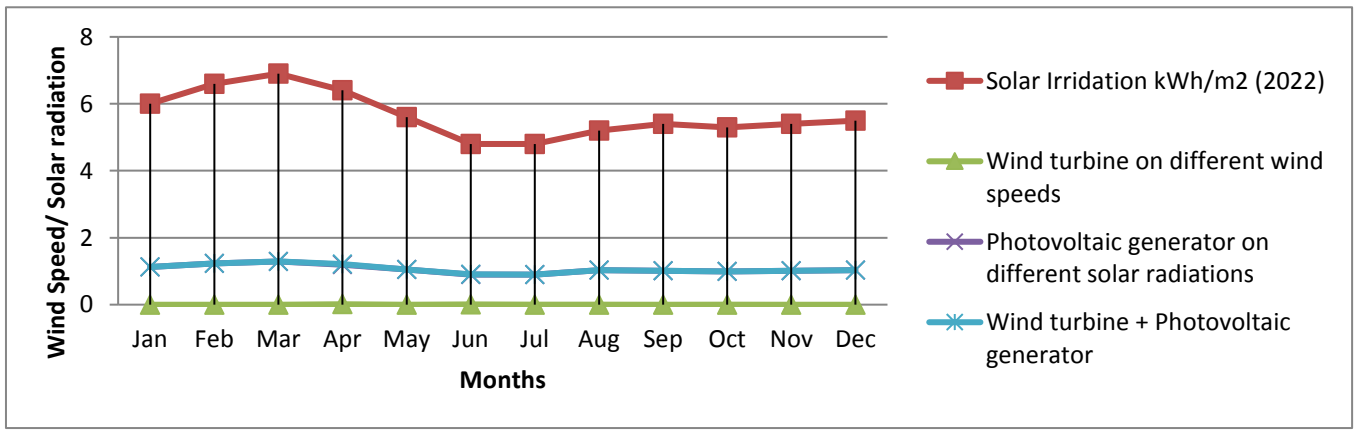


Fig. 10: Output value of wind generator, Photovoltaic, hybrid generator graph of Shimla (India) recorded between January to December 2022

3.7 Sixth location (Jammu, city, cold and sunny climate)

The city of Jammu has cold and sunny climate that was analyzed in this section. On the basis of average annual solar irradiation (5.0 kWh/m^2) and average annual wind speed was 0.3 m/s and the output electric power of photovoltaic generator was 1.3 W. Similarly, the output electric power of wind generator was 0.005 W and the output Photovoltaic-Wind Hybrid System (PWHS) was 1.30 W. It was also seen by the linear programming that size ratio of photovoltaic and wind system was not possible due to the low wind speed. Table 13 and 14 along with Fig. 11 and 12 represent the relevant data collected at Jammu.

Table 13: Average wind speed of Jammu (India) recorded by world weather online between January-December 2022

Months	Wind speed m/s
Jan	0.3
Feb	0.3
Mar	0.3
Apr	0.3
May	0.3
Jun	0.3
Jul	0.3
Aug	0.0
Sep	0.3
Oct	0.3
Nov	0.3
Dec	0.3

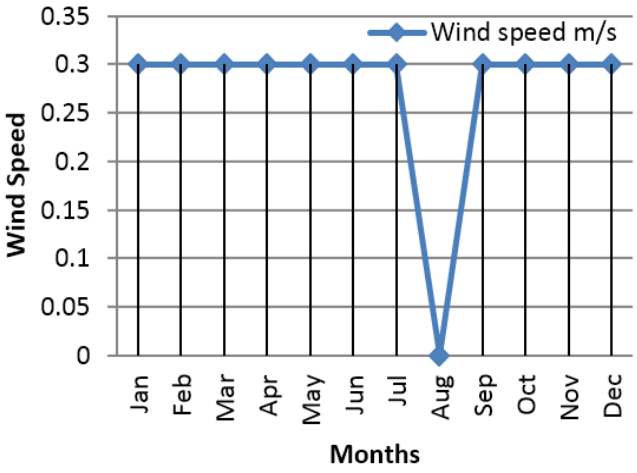


Fig. 11: Monthly average wind speed graph of Jammu (India) which was recorded between January to December 2022

Table 14: Output value of wind generator, Photovoltaic, hybrid generator (year 2022)

Months	Wind speed m/s	Solar Irridation kWh/m^2	Wnd turbine on different wind speeds	Photovoltaic generator on different solar radiations	Wind turbine + Photovoltaic generator
Jan	0.3	3.3	0.005	0.617	0.6175
Feb	0.3	4.1	0.005	0.766	0.7665
Mar	0.3	5.3	0.005	0.991	0.996
Apr	0.3	6.7	0.005	1.25	1.255
May	0.3	7.1	0.005	1.32	1.325
Jun	0.3	7.8	0.005	1.45	1.455
Jul	0.3	7.0	0.005	1.30	1.305
Aug	0.0	6.4	0.00	1.19	1.195
Sep	0.3	6.1	0.005	1.14	1.145
Oct	0.3	6.1	0.005	1.14	1.145
Nov	0.3	3.9	0.005	0.72	0.725
Dec	0.3	3.2	0.005	0.59	0.595

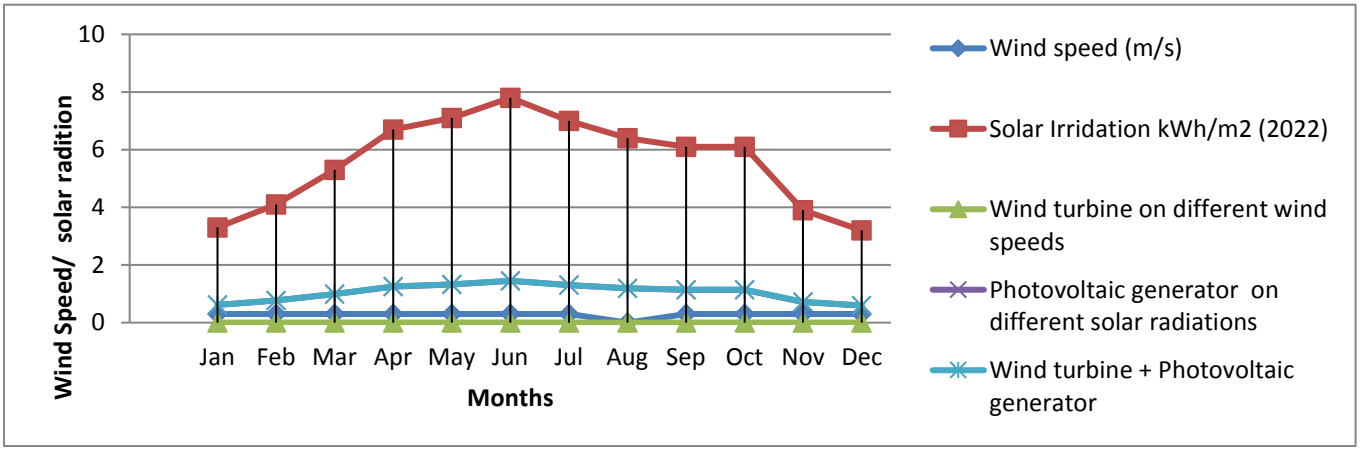


Fig. 12: Output value of wind generator, Photovoltaic, hybrid generator graph of Jammu (India) in 2022

Fig.13 shows the output values of wind generator, photovoltaic, hybrid generator of the first location (Indore) which is the composite climate. Second location is the hot and dry climate of Jaipur. The third location is the warm and humid climate of Dibrugarh and fourth location is the moderate climate of Bangalore, the city of Karnataka. Fifth location is the city of Shimla with cold and cloudy climate and sixth location is cold and sunny weather of Jammu. These values were recorded from January to December 2022.

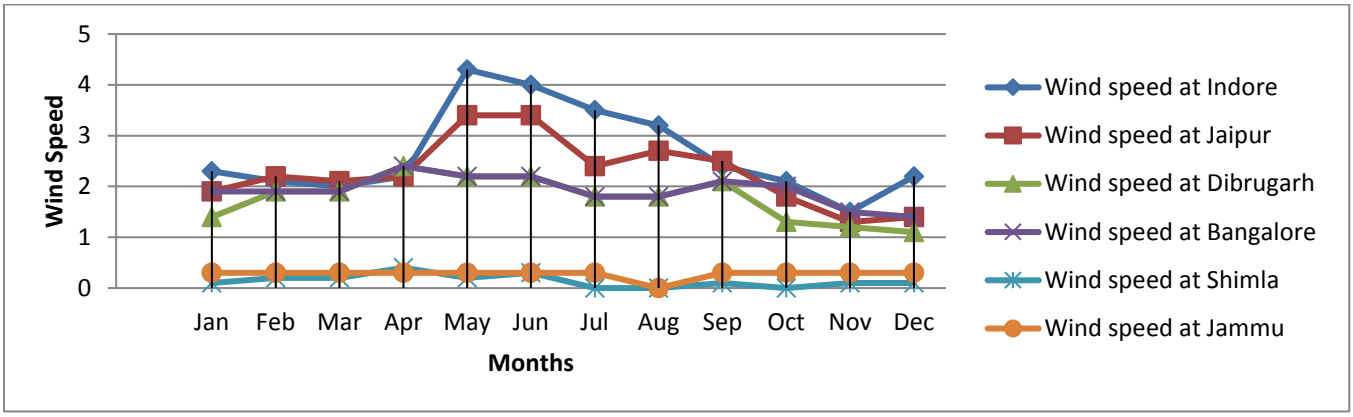


Fig. 13 Output values of wind generator, photovoltaic, hybrid generator graph at different locations of India

4. Result and Discussion

On analyzing Photovoltaic-Wind Hybrid Systems in different climatic zones, it was observed that the selected city of Indore with composite climate has suitable size ratio of PV wind that is 40:60. Jaipur with hot and dry climate having suitable size ratio of PV wind was 30:70. Dibrugarh possessing warm and humid climate with suitable size ratio of PV wind was 30:70. The photovoltaic output of Jammu with cold and is better than Shimla with cold and cloudy climate but output of the wind turbine is negligible over there because of the very low wind speed at both places. Therefore, the size ratio of PV wind was not determined.

5. Conclusion

Six different climactic locations of India were analyzed to find the size ratio of photovoltaic and wind PWHS on the basis of global solar irradiance and wind speed. Bangalore is said to be perfect location for PWHS with highest output in comparison with other locations of India. The size ratio of PWHS was 50:50 and it can be used for better understanding of the meteorological features characteristic of India. This study can help the domestics and industries to decide the location and size ratio of PV wind hybrid system for best utilization of the economy and availability of electrical energy. The result showed that solar radiation of Indore and Japur was 5.80 KWh/m^2 followed by Bangalore, Dibrugarh, and Shimla which was 5.6 KWh/m^2. On the other hand, the speed of wind at Indore was 2.4 m/s, Jaipur was 2.27 m/s, Bangalore was 1.8 m/s, Dibrugarh was 1.77 m/s, and wind speed at Jammu and Shimla were 0.3 m/s and 0.1 m/s, respectively. These locations are not recommendable for the wind energy production. If the energy production system was combined as photovoltaic wind hybrid system, we will get almost the same power output in Indore Bangalore that is followed by Jaipur and then Dibrugarh in accordance of the available data and calculations.

References

[1] C.R. Kumar. and M.A. Majid, "Renewable energy for sustainable development in India: current status, future prospects, challenges, employment, and investment opportunities," *Energy Sustain. Soc.*, vol. 10, no. 1, 2020.

[2] A.K. Das, "An evaluative study of some selected libraries in India undergoing the process of digitization*", Doctoral dissertation*. India, pp. 1–374, 2008.

[3] S.R. Sharvini, Z.Z. Noor, C.S. Chong, L.C. Stringer, and R.O. Yusuf, "Energy consumption trends and their linkages with renewable energy policies in East and Southeast Asian countries: Challenges and opportunities," *Sustain. Environ. Res.*, vol. 28, no. 6, pp. 257–266, 2018.

[4] S. Luthra, S. Kumar, R. Kharb, M.F. Ansari, and S.L. Shimmi, "Adoption of smart grid technologies: An analysis of interactions among barriers," *Renew. Sustain. Energy Rev.*, vol. 33, pp. 554–565, 2014.

[5] B. Bora, R. Kumar, O.S. Sastry, B. Prasad, S. Mondal, and A.K. Tripathi, "Energy rating estimation of PV module technologies for different climatic conditions," *Sol. Energy*, vol. 174, pp. 901–911, 2018.

[6] Y. Kwon, A. Kwasinski, and A. Kwasinski, "Solar irradiance forecast using naïve Bayes classifier based on publicly available weather forecasting variables," *Energies*, vol. 12, no. 8, pp. 1–13, 2019.

[7] B. Navothna and S. Thotakura, "Analysis on large-scale solar PV plant energy performance–loss–degradation in coastal climates of India," *Front. Energy Res.*, vol. 10, 2022.

[8] M. Bhatnagar, J. Mathur, V. Garg, and J. Iqbal, "Development of a method for selection of representative city in a climate zone," *Ashrae.org.*, vol. 8, pp. 510–517, 2018.

[9] A.K. Berwal, "Development of Statistical Model to Evaluate the Effect of Meteorological Parameters on the Performance of PV-Cell/Module," *Journal of Advanced Research in Alternative Energy, Environment and Ecology*, no. 1, pp. 1–7, 2020.

[10] A.S. Al-Ezzi and M.N.M. Ansari, "Photovoltaic solar cells: A review," *Appl. Syst. Innov.*, vol. 5, no. 4, p. 67, 2022.

[11] M.A.M. Ramli, A. Hiendro, and Y.A. Al-Turki, "Techno-economic energy analysis of wind/solar hybrid system: Case study for western coastal area of Saudi Arabia," *Renew. Energy*, vol. 91, no. C, pp. 374–385, 2016.

The Nucleus ***61***, *No. 1 (2024) 40-50*

www.thenucleuspak.org.pk

The Nucleus
ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Balancing Energy Consumption and Connectivity through Energy-Efficient Protocols for Sustainable Networking in Smart Cities

Uroosa Bilal Chaudhry[1*], Tauqir Ahmed[2], Amna Wajid[1]

[1]*Department of Computer Science, Rachna College of Engineering and Technology, Gujranwala, Pakistan*

[2]*Department of Computer Science, University of Engineering and Technology, Lahore, Pakistan*

ABSTRACT

This paper focuses on the challenges and opportunities of achieving sustainability in smart cities. It depends on maintaining the balance between energy efficiency and connectivity through energy-efficient protocols. The study surveys systems with energy-efficient protocols, emphasizing their strengths and limitations in maintaining sustainability in smart cities. The need for energy-efficient protocols in smart city settings is that these protocols dynamically adjust operations to optimize energy consumption within sensors nosed in communication networks while considering parameters like heterogeneity, traffic patterns, and environmental factors. Evaluation against key performance metrics such as energy consumption, data delivery ratio, network lifetime, and throughput identify promising technologies to achieve an optimal balance between connectivity and energy efficiency in smart cities. It concludes by proposing future research directions and considerations to make the development of sustainable networking solutions rapidly tailored to the evolving landscape of smart cities. This paper aims to present a comprehensive overview of the pivotal role played by energy-efficient protocols in fostering sustainable networking architectures.

Keywords: *Green IoT, Wireless sensor networks, Energy-efficient protocols, Smart cities, Machine Learning*

1. Introduction

In the dynamic landscape of urban development, the emergence of smart cities is a beacon of innovation, promising enhanced quality of life, resource optimization, and efficient service delivery. At the heart of these interconnected urban hubs lies a complex web of networking infrastructures, powering the seamless operation of diverse applications and services. Fig. 1. illustrates the basic smart city infrastructure. However, pursuing sustainable networking in these cities presents a pressing challenge: the delicate balance between energy efficiency and robust connectivity [1].

In recent years, several cities around the world have continued to implement innovative smart city initiatives aimed at enhancing energy efficiency and sustainability. For example, Copenhagen's ambitious goal to become the world's first carbon-neutral capital by 2025 has spurred numerous smart city projects. One notable initiative is the CityFlow project, which uses advanced data analytics and sensor technologies to optimize traffic flow and reduce congestion in the city center. By dynamically adjusting traffic signals based on real-time data, Copenhagen aims to minimize energy consumption and emissions from vehicles [3].

Smart cities, by their very nature, rely on interconnected systems fueled by data-driven technologies. From the Internet of Things (IoT) devices facilitating real-time information exchange to intelligent traffic management systems optimizing transportation, these networks underpin the functionality of urban ecosystems. Yet, this interconnectivity comes at a cost—significant energy consumption and the subsequent environmental impact.

Efforts to curtail energy usage often clash with the demand for uninterrupted connectivity, creating a dilemma that necessitates a paradigm shift in networking protocols.

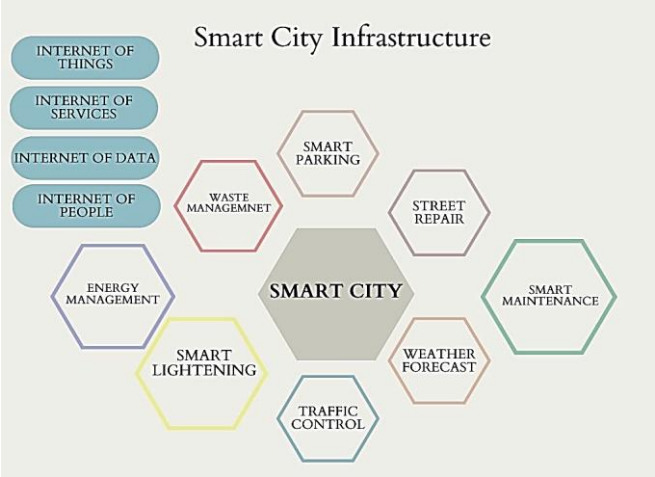


Fig. 1. Smart city's infrastructure.

The imperative lies in devising adaptive protocols that harmonize energy-efficient practices with the imperatives of sustaining high-level connectivity. These protocols dynamically adjust network behaviors, optimizing energy consumption while ensuring reliable connections for critical applications [2].

This paper delves into the critical juncture where sustainability, technological innovation, and urban development intersect. It seeks to explore the challenges posed by energy-intensive networking infrastructures in smart cities and propose adaptive protocols as a viable solution to achieve an equilibrium between energy efficiency and connectivity. By examining the significance of energy-efficient networking and the role of adaptive protocols in reconciling conflicting objectives, this paper aims to provide insights into sustainable networking practices tailored for the smart cities of tomorrow.

*Corresponding author: uroosa.bilal@uet.edu.pk

This research paper presents a novel approach to addressing the energy efficiency and connectivity dilemma in smart cities through the development and implementation of adaptive networking protocols. Unlike existing studies that primarily focus on theoretical frameworks or individual case studies, this paper offers a comprehensive analysis of sustainable networking practices, specifically tailored for the complex and dynamic environments of smart cities. Additionally, the paper introduces a systematic methodology for evaluating the effectiveness of adaptive protocols in real-world urban settings, providing valuable insights for policymakers, urban planners, and technology developers. Through its interdisciplinary approach and practical recommendations, this paper contributes significantly to the existing body of knowledge on sustainable networking in smart cities, paving the way for a more sustainable and connected urban future.

The organization of this research paper is given as follows. The first section is an introduction that provides an overview of the challenges and opportunities in sustainable networking for smart cities and highlights the necessity for adaptive protocols. The second section is a literature review, in this section, a detailed review of sustainable smart cities is presented in this paper, these studies have been evaluated on networking mode between communication devices, technologies adopted to control energy consumption, and AI/ML technologies to provide a comprehensive solution to maintain sustainability in smart cities. After analyzing all these scenarios from a broader spectrum, we have narrowed down the communication technology to only wireless sensor networks, and protocols to energy-efficient protocols to embed them with AI mechanisms to provide an energy-efficient solution and produce better results than the conventional mechanism. After that systems with heterogeneous energy efficient protocols are reported and analyzed on chosen parameters like energy consumption, data delivery ratio, network lifetime, and throughput to give a deep insight into their strengths and limitations. After that research objectives and research novelty are discussed. The main section is dedicated to protocol utilization in the communication of devices for maintaining sustainability in smart cities. After this evaluation discussions are available. In the end, this paper has concluded with future directions to motivate researchers to contribute and explore this domain further.

2. Literature Review

The IoT network necessitates the interconnection and establishment of communication links among a vast and diverse array of ubiquitous devices, resulting in the handling of substantial amounts of data. This is particularly challenging due to limitations in computing capacity and battery life, as well as other factors. Another aspect to consider is the environmental cost, of the global technological advancements that have occurred over the past few decades. The proliferation of billions of IoT devices necessitates a significant amount of energy, as these devices generate massive amounts of data that traverse networks. Additionally, the billions of batteries used to support these IoT devices are discarded and needed throughout their life cycle. The reviews based on sustainability measurements and measures for improvement are being analyzed to identify the gap in a specific domain.

Mohanty has provided an introductory review of smart cities and explained their working. Traditional networks work more flexibly with the services and provide a sustainable environment. Smart cities provide smart infrastructure, energy, health care, and technology. Transforming traditional cities into smart cities highlights the importance of information and communication technology. These standards can help in maintaining growth, quality, efficiency, and most importantly safety [1]. Challenges in designing smart cities are merely discussed in the paper which offers opportunities to work in cost maintenance, efficiency building, communication, and security.

Angelidou and colleagues have delved into the capacity of smart city methodologies and tools in advancing sustainable urban development within the environmental sphere. Emphasizing the necessity for a more methodical inquiry into the interplay between smart and sustainable cities, the study predominantly centers on practical implementations. The research scrutinizes 32 smart city applications sourced from the Intelligent/Smart Cities Open Source (ICOS) community, specifically focusing on their relevance to environmental sustainability. The comprehensive analysis encompasses various elements such as addressed environmental concerns, strategies for mitigation, innovative mechanisms, the role of information and communication technologies, and the overarching outcomes. The outcomes highlight a fractured landscape in both policy and technical dimensions of smart and sustainable cities. The study uncovers numerous unexplored prospects for smart sustainable development that remain largely undiscovered. This fragmentation is observed across all categories of environmental challenges in urban settings. However, it's crucial to acknowledge the study's limitations due to the relatively small number of applications examined. Nevertheless, the research underscores the potential for smart city applications to significantly influence sustainable urban development. The implications of these findings extend to policymakers, offering insights to bolster their proactive role on local and global stages. Furthermore, the delineation of market niches for smart city applications holds significance for developers, user communities, and digital entrepreneurs [13].

Alsamhi, Saeed has delved into the concept that in the effort to make cities smart, greener (through the concept of green IoT), and sustainable, there are surely many challenges that need to be tackled. Carbon footprint, energy consumption, and pollution are the main challenges when sustainable infrastructures are discussed. Green cities should be eco-friendly for faster and more comfortable adaption. This paper provides a comparative review of many strategies that aim to attain sustainable eco-friendly smart cities. Integration of IoT in many aspects like e-healthcare, home automation, industry digitization, and transportation autonomy are

discussed. The main focus is to present IoT as a sustainable solution for maintaining public safety, energy efficiency, and minimizing pollution. Further, this paper elaborates that combining AI with green ICT principles can be a very effective solution to decrease energy consumption and cutoff carbon emissions [3].

Bruneo, Dario emphasizes that utilizing open technologies is the ultimate solution for transforming a city into a smart city. This paper discusses an initiative in Italy to develop a smart city with great IoT technologies. Affordable sensor devices and intelligent services have been implanted to achieve the goals of green IoT with the Stack4Things framework. This paper [4] in particular discusses the challenges of integration in implanting the project and some specialized solutions for continuous monitoring of the environment and efficient management of parking facilities.

Alsamhi, Saeed H., et al. delve into the integration of IOT and the potential challenges associated with it. Besides the numerous benefits of IOT, many challenges are also embedded with it. It mainly includes pollution, e-waste generation, and energy consumption. To minimize carbon emissions and the generation of e-waste, the concept of "green IoT" is widely adapted to build sustainable cities. This paper provides a comprehensive review of potential technologies related to green IoT. [16].

Pattnayak, et al. highlight that cloud service with the combination of high-speed broadband networks provides potential research areas for further investigation. Simultaneously, sensor networks employing various wireless technologies in eco-friendly smart cities provide access to information regarding the flow of goods, equipment status, and environmental conditions. These networks also enable remote control, promoting the development of environmentally conscious smart cities that prioritize safety. Collaboration among these networks is encouraged as sensors, communication technologies, and control systems become more advanced and interconnected. Enabled by the Internet of Things (IoT), big data applications are becoming increasingly essential for smart sustainable cities' operational functioning and planning, contributing to their environmentally sustainable development goals [7].

Maccani, Giovanni, and colleagues have focused on achieving harmony in the implementation of IT government (ITG) and incorporating smart city functions (SCF) with the help of local authorities. The paper focuses on the adaptation of the ITG mechanism in local governments and this case study is conducted in Irish cities' authorities. These types of structural mechanisms can be adapted in ITG. Integrated, Detached, and Traditional. Comparison is made based on parameters like decision-making authority, alignment with IT function, and overall governance. These observations can help make the decisions related to the adaptive challenges faced by SCFs [8].

Ahad, Mohd Abdul, et al. discuss that enabling technologies in smart cities is the most important factor in designing their infrastructure and providing facilities to comfort the users. This paper summarizes many frameworks and research that have been conducted in this area. Attaining sustainability is the biggest challenge in developing smart cities. The authors discuss the main frameworks and technologies to build the architecture of smart cities while maintaining sustainability. The Internet of Things (IoT) provides a foundation for all of this. Still, the most concerning areas are providing security and privacy within the sustainable environment. For this, this paper provides a comparison of security protocols and their implementation to give an insight into security frameworks in smart cities [2].

Ahad, Mohd Abdul, and colleagues extensively review the potential and limitations of enabling technologies in smart cities. It provides extended examples and categorizes the challenges in three categorical areas: technical, socio-economic, and environmental. This paper summarizes the best practices for building sustainable smart cities [12].

Verma et al. have commenced by presenting an in-depth exploration of energy management within Green IoT, highlighting the challenges stemming from the extensive utilization of key IoT devices. The study extensively examines diverse approaches aimed at reducing energy consumption in IoT ecosystems. These include the optimization of data centers for enhanced efficiency, the implementation of energy-conscious data transmission from sensors, and the innovation of energy-efficient solutions. Acknowledging the constrained energy sources powering IoT devices, such as batteries, the paper delves into addressing energy efficiency concerns across various IoT networks. It introduces multiple framework models for Green IoT, alongside proposing energy-efficient designs to elongate the operational lifespan of IoT devices.

Moreover, the paper scrutinizes strategies for green IoT, offering recommendations to realize objectives in this domain. Additionally, it undertakes a comparative analysis considering techniques, classifications, constraints, and other pertinent factors. This endeavor aids researchers in acquiring comprehensive insights and refining existing methodologies for superior energy management in forthcoming practices [15].

Akin-Ponnle, et al. have explored the crucial challenge of powering Internet of Things (IoT) nodes within smart cities using sustainable energy sources amid the COVID-19 pandemic's impact on remote interactions. The focus lies on energy harvesting as a viable solution for enabling IoT devices to harness energy from diverse environmental reservoirs. A comprehensive review is conducted of potential energy sources available within urban settings, drawing upon existing literature. By highlighting the significance of aligning

Table 1: A parametric analysis of previous works in green IOT

Ref	Major Contribution	RFID	WSN	M2M	Energy harvesting	Carbon Footprint	Social Impact	AI/ML	Recycling/ policies
[1]	Discusses the different components of a smart city, including smart infrastructure, smart transportation, smart energy, smart healthcare, and smart technology	✓	✓	✓	×	×	×	×	×
[2]	Emphasizes the importance of sustainability in smart city development and highlights the challenges and best practices associated with it	×	×	×	✓	✓	✓	×	×
[3]	Explores the integration of IoT technologies into various aspects of smart cities, including environmental monitoring, e-healthcare, transportation autonomy, industry digitalization, and home automation	×	×	×	✓	✓	✓	✓	✓
[4]	Strategically placing affordable sensor-powered devices throughout the urban landscape, a diverse range of intelligent services has been implemented	✓	✓	✓	×	×	✓	✓	✓
[5]	Green IT concepts to create sustainable and eco-friendly public services	×	×	×	✓	✓	✓	✓	✓
[6]	Highlights that the proliferation of IoT devices has led to increased energy consumption, electronic waste (e-waste) production, and carbon emissions	✓	✓	✓	✓	✓	✓	×	×
[7]	Sensor networks employing various wireless technologies in eco-friendly smart cities provide access to information regarding the flow of goods, equipment status, and environmental conditions	✓	✓	✓	✓	✓	✓	×	×
[8]	Three types of ITG structural implementation are identified: Detached, Integrated, and Traditional. These are compared in terms of their orientation, decision-making authority, alignment with the IT function, and the overall municipal organization	✓	✓	✓	✓	✓	×	×	×
[9]	Need for efficient computing solutions to reduce service response times in IoT applications.	×	×	×	✓	✓	✓	✓	✓
[10]	Demonstrates the effectiveness of using neuro-fuzzy logic for managing resources and creating dynamic clusters in sustainable IoT devices for green smart city applications	✓	✓	✓	×	×	✓	✓	✓
[11]	Computational optimization algorithms have proven effective in optimizing energy consumption and reducing pollution and greenhouse gas emissions	✓	✓	✓	×	×	✓	✓	✓
[12]	Categories challenges into three main areas: technical, socio-economic, and environmental, providing specific examples within each category	✓	✓	✓	✓	×	×	✓	✓
[13]	The identification of market niches for smart city applications is of interest to developers, user communities, and digital entrepreneurs	×	×	×	✓	✓	✓	✓	✓
[14]	Emphasize the importance of tailoring energy sources to specific applications and suggest that energy scavenging should occur in proximity to where it is needed for IoT devices and wireless sensor networks, ultimately contributing to the automation and optimization of smart cities	✓	✓	✓	×	×	✓	✓	✓
[15]	A comparative study based on techniques, types, limitations, and other factors to assist researchers in gaining insights into this domain and improving current practices for better energy management in the future	×	×	×	✓	✓	✓	✓	✓
[16]	The study emphasizes the need for machines, sensors, communication networks, clouds, and the internet to work harmoniously to enhance energy efficiency and mitigate carbon emissions.	✓	✓	✓	×	×	✓	✓	✓

energy sources with specific applications and advocating for localized energy scavenging to support the energy needs of IoT devices and wireless sensor networks, this localization is projected to play a pivotal role in the automation and optimization of smart cities [14].

Hasmawaty has focused on the potential of ICT technologies to transform public services into sustainable solutions to protect the environment. The green smart cities combine the concept of smart cities with green IoT, in which we adapt technology with sustainable environment solutions to reduce energy consumption. This paper takes the case study of the South Sumatra government and their adoption of green IoT in developing a greener smart city. The findings of this case study highlight the many sustainable solutions for achieving green IoT with controlled carbon emissions and lesser pollution. This study has provided many sustainable solutions and discussions for achieving sustainable growth by adapting the smart cities concept [5].

He, Ping, and colleagues have embraced the fact that energy efficiency is a crucial need for a sustainable environment and a healthy society. Energy resource management and optimizing energy resource management should be the main focus of researchers to work on and draw mechanisms to achieve a sustainable and efficient environment with modern technologies embedded in smart cities. Cloud computing and IoT are enhancing the advancement in smart cities, making people's lives easier and more comfortable. This paper provides comprehensive research in this area to enlighten the importance of incorporating energy consumption optimization and efficiency to reduce pollution and greenhouse gas emissions in smart cities [11].

Albreem, Mahmoud A., has emphasized that with evolving technologies in Gulf countries, an energy crisis is arising. IoT technology benefits the people but it also damages the environment by producing massive CO_2 emissions. Energy consumption waste and CO_2 emissions are the massive challenges that the cities have to face in return for embedding technologies to convert their traditional cities into smart cities. Adapting green IoT is the solution to cut these damages down. The main aim of green IoT is to reduce the E-waste and CO_2 emissions due to increased technology adaptation these days. This paper mainly focuses on the challenges in adapting green IoT and strategies that can be embodied to overcome these challenges. It also gives insights into the government policies to overcome these challenges. Overall, this paper summarizes the challenges in adaption GioT and strategies that have been taken by the government especially of Gulf countries to reduce the consequences of incorporating technology into lives.

Khanh, et al. highlight the evolution of technology and the emergence of 5G technology which has revolutionized the connectivity in IoT scenarios and increased their performances dramatically. 5G technology changes the connectivity of IOT for the better in health, agriculture, education, military, and smart cities because it can make the IoT network work more rapidly than ever before and the number of devices that the technology can support, is also humungous. The main challenge addressed in the study is the need for efficient computing solutions to reduce service response times in IoT applications. An edge computing mechanism was proposed to facilitate IoT technology in smart cities. A small database is established known as an “information map” which allows edge computing to manipulate and store information in smart cities and is utilized to transfer information within the edges. The results show that the proposed mechanism reduces energy consumption and CO2 emissions as compared to previous solutions. This paper concluded with the vision that the proposed system can be expanded to be embedded in smart cities in the future [9].

Chitahur’s study has elaborated on the significance of the Internet of Things (IoT) for fostering sustainable smart city applications, each tailored to distinct necessities like energy efficiency, Quality of Service (QoS), and resource management. They present an energy-conscious Dynamic Clustering Routing (DCR) protocol employing a neuro-fuzzy approach to tackle resource limitations within IoT devices. This protocol leverages a dynamic self-organizing neural network for cluster creation within the network. The research encompasses a test-bed analysis that encompasses real-time event identification and sensor node clustering through TinyOS.

3. Research Novelty

Table 1 represents the comparative analysis of the previous review. Review [1] focuses on the communication modes. Some reviews are focused on communication modes and carbon footprint emissions [6-8]. Other reviews are based on energy consumption, CO2 footprints, and AI/ML-based techniques to make IoT mechanisms energy efficient [3] [5] [9] [13] [15]. While other reviews mainly focus on communication modes and AI/ML-based techniques to adopt green IoT [4] [10] [11] [12] [14] [16]. There is not a single review that covers wireless sensor networks with energy-efficient protocols, energy consumption, and AI/ML-based mechanisms for maintaining sustainability in smart cities. This paper aims to combine all of these aspects in a single repository to give researchers an insight into important aspects related to achieving energy-efficient communication in smart cities.

4. Research objective

The research objectives of this review are

- To report the previous reviews done by researchers on the sustainability of green IoT.
- To give a comparative analysis of previous reviews on chosen parameters.
- To find a gap in previous work and build our review on that.
- To propose a methodology to present our review of systems that combine wireless sensor networks, energy-efficient protocols, and AI/ML mechanisms to give a comprehensive solution to energy efficiency and maintainability in smart cities.

5. Protocols for Sustainable Smart Cities

In smart city deployments, selecting the appropriate communication protocol depends on factors such as power consumption, range, bandwidth, scalability, security, and

interoperability requirements. Often, a combination of these protocols is used within the same ecosystem to ensure comprehensive coverage and effective communication between various devices and systems while maintaining sustainability.

Dbibih, has proposed an algorithm BMPriority-based CSMA/CA, based on battery energy level and message priority of sensors for accessing transmission channels with better precision. The proposed solution utilized a weighing function for window contention calculations, while the sensor node waits before starting the transmission. The authors claimed that the proposed algorithm protocol outperforms S-MAC, IEEE 802.15.4, and ECA-MAC protocols in terms of optimizing Packet Delivery Ratio (PDR), end-to-end latency, and throughput [22].

Rani, N, et al., have demonstrated a cluster-based routing algorithm that uses different schemes to make energy consumption efficient. The high energy level nodes in the routing mechanism are used for transmissions and others are used to sense data. This phenomenon is known as cluster-based routing which is used to transmit data at the fastest rate. Hierarchical protocols like LEACH, HEED, TEEN, SEP are used in the proposed solution [23].

Nadia has proposed an energy-efficient system based on reinforcement learning. The main protocols that are focused on in this solution are IEEE 802.15.4 and the Medium Access Control (MAC) to attain the minimum energy consumption through communication nodes. MAC protocol is used to compute the duty cycle that ensures optimal performance and minimum energy consumption in communication modes in IoT. The combination of reinforcement learning with energy-efficient protocol adaptation provides energy sustainability and Quality of Service (QoS) [24].

Rana, A, K. et al. have elaborated an Advanced Zone-Stable Election Protocol (AZ-SEP). Some information is transferred from the base station. The bunching method is utilized to transfer data to the base station. The proposed adaptive protocol is compared with the clustering hierarchy (LEACH) and it outperformed the conventional protocols by 64% and the simulation results have shown enhanced throughput, data aggregation, and minimal energy consumption [25].

Chithaluru, P., has proposed Improved-Adaptive Ranking based Energy-efficient Opportunistic Routing protocol (I-AREOR). It is based on relative distance, residual energy, and regional density. It mainly focuses on a clustering approach that clusters the nodes into three categories, First-node death, half-node death, and last-node death. The proposed system gives a solution to the challenging task of extending the time of first node death by utilizing the regional density and residual energy of nodes. This system considers the energy parameters to provide optimal energy consumption between nodes' communication [26].

Haseeb, K., Din, et al., have demonstrated a secure and intelligent model for sustainable cities using green IoT and it mainly functions on edge computing to achieve lower energy consumption rates. Deep learning has been utilized for data routing. Furthermore, distributed hashing is implemented with a chaining strategy to provide a secure as well as efficient system. These proposed systems outperformed the previous systems in terms of energy consumption, and network throughput [27].

Mishra, M, and colleagues have elaborated on the importance of clustering mechanisms to group the nodes into clusters. The optimal energy consumption leads to the maximum network lifetime. In this paper, a multi-objective optimization approach is proposed to select the optimal route for data transmission among the transmitting nodes. The proposed system is a two-step approach. The first step is detecting cluster heads for data communication within nodes in clusters. Secondly, the combination of a particle swarm optimization (PSO) algorithm and a genetic algorithm (GA) is proposed to determine the optimal routing path between nodes. LEACH protocol I utilized in the proposed system. The simulations show that this system outperforms the conventional systems in terms of energy consumption and network lifetime [28].

Alvi, A. N., has focused on an Optimal GTS Allocation Mechanism with an Adaptive Duty cycle (OGMAD). IEEE 802.15.4 standard is utilized to focus on low data rate communications within the sensor nodes. It adapts the optimal ‘guaranteed Time Slot” mechanism to improve the link utilization within communication nodes. The simulation results showed that the proposed system provides better duty cycles in the network and optimal energy consumption [29].

Abdul-Qawy, A. S. H., and colleagues have proposed a scalable and energy-efficient system for wireless nodes in green IoT. It includes 3 main components. 1. Hybrid placement scheme 2. A heuristic model for multi-stage weighted election heuristic (MSWE) 3. Cross-layer transmission model for minimum cost management. This model achieves a minimum energy cost from the bottom layer and moves towards to topmost layer for data transmission. The simulation results show that the proposed scheme is better than the conventional protocols regarding energy consumption, network throughput ns stability [30].

Sodhro, A. H., has demonstrated a Hybrid Adaptive Bandwidth and Power Algorithm (HABPA), and Delay-tolerant Streaming Algorithm (DSA). This system is designed to control the power loss, packet loss ratio, and standard deviation. It is obvious from the simulations that the proposed system is an efficient approach for energy consumption within the communication nodes. This perspective enlightens the importance of sustainability in smart cities [31].

Alharbi, M. A., has proposed a system that focuses on the combination of the clustering approach with the routing protocol to give a comprehensive solution for energy efficiency within communication nodes. The system relies on the area-based clustering that is derived from the transmission rate with the nodes. Cluster heads are chosen to avoid failure in the routing paths for finding the optimal routing path hop-count is utilized. The results of experimentation show that the system provides improved network lifetime, energy efficiency, and improved network density [32].

Iala, I., and colleagues have developed an effective and energy-conscious protocol that is better in performance than the conventional protocols in terms of energy consumption and network lifespan. This paper proposed a new mechanism based on 802.15.4 wireless sensor nodes called Kalman filter-based MAC protocol. It is designed to optimize the energy consumption on nodes. It extends the life span of network nodes. This new protocol optimizes the sleep intervals that happen on the

network nodes. The awakening of sensor nodes should happen only when it is needed. It is a postponing-based transmission mechanism that resolves the network collision problem. This mechanism produces better results in terms of energy efficiency and packet delivery ratio [33].

Deepa, O., has proposed Optimized QoS-based Clustering with Multipath Routing Protocol (OQoS-CMRP) for wireless sensors to reduce energy consumption in wireless sensor nodes. Particle Swarm Optimization (PSO)-based clustering algorithm is applied to create clusters. To find the neighbor nodes The Single Sink-All Destination algorithm is utilized. The performance of the proposed system outperformed the conventional protocols in terms of transmission delay, communication overhead, and energy efficiency [34].

Maheshwar's study has focused on the clustering and routing algorithm to enhance the network lifetime and minimize the energy consumption within sensor nodes. Butterfly Optimization Algorithm (BOA) is utilized to create a cluster head and Ant Colony Optimization (ACO) is utilized to identify the base station. The mechanism identifies the optimal route based on node degree, residual energy, and distance. The performance metrics of this mechanism are alive and dead nodes, energy consumption, and data packet ratio [35].

Rami Reddy, and colleagues have proposed a mechanism that is bad for energy recovery nodes in the network to propose an optimal energy consumption algorithm. The combination of Fuzzy-GWO method and energy-efficient routing algorithm is proposed to overcome the network overhead in the most energy-efficient way. The simulations have shown that the proposed mechanism is better than conventional protocols regarding data packet delivery ratio, network throughput, and energy consumption [36].

Sadek, R. A., has elaborated on an energy-efficient protocol that is based on the combination of the grey wolf optimizer (GWO) and the Tabuse arch algorithm (TSA). this mechanism works on the clustering approach. The clustering heads are identified by utilizing GWO based on the residual energy of nodes and the average distance between the cluster head and the sink nodes. The proposed mechanism enhances the quality of service (QoS) which involves reliability and energy consumption. The simulations show that the proposed system outperforms the conventional protocols regarding network lifetime, throughput, and energy consumption [37].

Wang, Z., has proposed an improved artificial bee colony algorithm (ABC) that focuses on the clustering of network nodes. By utilizing this algorithm, cluster heads are identified based on cluster head energy, density, and location. This mechanism optimizes the clustering process across the network nodes. The combination of the ABC algorithm with Fuzzy C-mean clustering provides an optimal clustering method to route between the cluster head and the base station. The experimental results have shown that the proposed mechanism is better than the conventional protocols regarding network throughput and energy consumption efficiency [38].

Elhoseny's study demonstrates a novel swarm intelligent-based clustering and the multi-hop routing protocol to cluster the network nodes more proficiently. The grey wolf optimization algorithm (GWO) is utilized to identify the optimal paths in networks. the selection of cluster heads and their organization is the initial step. then the GWO algorithm is applied. this mechanism incorporates both clustering and the routing processes. This results in the better network throughput and energy consumption efficiency. The simulations show that the proposed mechanism outperforms the conventional protocols [39].

Sahoo, B. M., has proposed a mechanism based on the clustering approach and a combination of genetic algorithm (GA) and particle swarm approach to provide optimal network performance. Cluster head is identified by utilizing the GA algorithm. GA helps in identifying the cluster head proficiently and PSO helps in finding the optimal route within the sensor nodes of the network. The experimental results show that the proposed system is better in network throughput and energy consumption efficiency [40].

6. Evaluation and discussion:

Evaluation of the various methods reveals their respective strengths and weaknesses. By analyzing their performance, we gain insights into their suitability for enhancing the sustainability of smart cities.

6.1 Analysis of existing approaches:

Table 2 depicts a diverse array of AI/ML techniques and clustering algorithms applied to improve routing protocols in wireless networks. Each referenced protocol introduces distinct methods to optimize network performance in terms of routing efficiency, energy consumption, data packet ratio, and network lifetime.

Reference [22] implements BMPriority-based CSMA/CA using clustering techniques, focusing on optimizing network throughput and energy consumption efficiency. [23] utilizes Fuzzy C-means clustering, aiming to enhance routing protocols, although specific performance metrics are not detailed. Meanwhile, [24] applies reinforcement learning on IEEE 802.15.4 and MAC protocols, potentially offering adaptive routing solutions. [25] presents the Advanced Zone-Stable Election Protocol (AZ-SEP), emphasizing stability in cluster-based routing.

Deep learning is leveraged in [27] to optimize MAC protocols, while [30] proposes a heuristic model using deep learning for weighted election heuristics, aiming for efficient routing. [28] combines particle swarm optimization and genetic algorithms for clustering, possibly indicating a robust optimization approach. [34] introduces the Multipath Routing Protocol (OQoS-CMRP) to enhance network lifetime and data packet ratio, focusing on multipath strategies.

6.1.1 Reduced Carbon Footprints

Table 2 outlines various technologies, performance metrics, AI/ML techniques, routing protocols, and their associated characteristics in the context of energy-efficient protocols for smart city networking. The following parameters influence the reduction of carbon footprints within smart cities.

Technology Used: The table lists a variety of clustering algorithms and routing protocols utilized in smart city networking. These technologies play a crucial role in optimizing energy consumption and reducing carbon footprints by efficiently managing data transmission and network operations.

Performance Metrics: The performance metrics outlined in the table, such as network throughput, network lifetime, energy consumption efficiency, and data packet ratio, are directly related to energy efficiency and carbon footprint reduction. Optimizing these metrics through the use of energy-efficient protocols can lead to significant reductions in energy consumption and associated carbon emissions.

AI/ML Techniques: The incorporation of AI/ML techniques, such as deep learning and reinforcement learning, enhances the capabilities of energy-efficient protocols by enabling adaptive and intelligent decision-making. These techniques contribute to optimizing network performance and reducing energy consumption, thereby mitigating carbon footprints.

Routing Protocols: The choice of routing protocols plays a critical role in determining the energy efficiency of smart city networks. By selecting routing protocols specifically designed for energy efficiency, such as those listed in the table, cities can minimize unnecessary energy expenditure and reduce carbon emissions.

Outperforms LEACH, HEED, TEEN, SEP Protocols: Several protocols in the table are noted to outperform traditional protocols such as LEACH, HEED, TEEN, and SEP in terms of energy efficiency and network performance. By adopting these advanced protocols, smart cities can achieve significant reductions in energy consumption and carbon footprints compared to conventional approaches.

To summarize all these findings, the technologies and protocols outlined in Table 2 offer valuable insights and tools for controlling carbon footprints in smart cities. By leveraging energy-efficient protocols, incorporating AI/ML techniques, and optimizing network performance metrics, cities can make substantial strides towards environmental sustainability and combatting climate change.

6.2. Limitations

However, limitations persist across these approaches. While the mentioned protocols target various performance metrics, a lack of comprehensive comparative analysis across all metrics exists. Real-world validation is also limited or absent, hindering a full understanding of practical feasibility and performance.

6.2.1 Field Trials in Urban Environments

Conducting field trials in actual urban settings provides an invaluable opportunity to validate the performance of energy-efficient protocols in real-world scenarios. A collaboration between researchers, city authorities, infrastructure providers, and industry partners to deploy prototype networks and IoT devices in urban areas is crucial to analyze network performance on real-time data. By collecting data on network performance, energy consumption, and environmental conditions over an extended period, the practical feasibility and effectiveness of the protocols can be assessed in diverse urban environments. Field trials also allow researchers to identify and address challenges such as signal interference, network congestion, and environmental factors that may impact protocol performance.

6.2.2 Simulation of Various Network Conditions

In addition to field trials, simulations offer a cost-effective and scalable approach to validate energy-efficient protocols under various network conditions. Researchers can leverage simulation tools and platforms to model different urban environments, network topologies, and traffic patterns. By adjusting parameters such as node density, communication range, and mobility patterns, researchers can evaluate protocol performance across a wide range of scenarios. Simulation studies enable researchers to systematically explore the impact of different factors on protocol performance, identify potential bottlenecks or vulnerabilities, and optimize protocol design accordingly.

6.2.3 Collaboration with Industry Partners

Collaborating with industry partners and stakeholders can facilitate the deployment and validation of energy-efficient protocols in real-world settings. Researchers can work closely with companies specializing in IoT devices, network infrastructure, and smart city solutions to integrate prototype protocols into existing systems or deploy pilot projects in urban areas. Industry partners can provide access to resources, expertise, and infrastructure needed for large-scale deployments, while researchers can contribute domain knowledge, technical expertise, and research insights. Collaborative efforts enable researchers to validate protocols in diverse operational environments, gather feedback from end-users, and iteratively refine protocol design based on real-world experiences and observations.

Moreover, scalability, adaptability, and computational complexity remain potential hurdles, impacting their deployment in diverse and dynamic network conditions.

6.3 Evaluation Metrics

Specific performance metrics like network throughput, lifetime, energy consumption, and data packet ratio have been mentioned, but a comparative analysis across all these metrics for each protocol might be lacking.

6.3.1 Real-world Validation

Limited or no real-world deployment and validation of these protocols could be a major drawback, hindering the understanding of their practical feasibility and performance.

6.3.2 Scalability and Adaptability

Some protocols might lack scalability or adaptability in dynamic network conditions, limiting their application in varying environments.

6.3.3 Complexity

Implementing AI/ML techniques or complex algorithms can result in increased computational overhead, affecting the feasibility of these protocols in resource-constrained devices.

7. Future directions

Based on the limitations identified in the existing protocols for wireless network optimization, several future directions could be pursued to advance the field:

7.1 Comprehensive Multi-Metric Evaluation

For future, researchers could focus on developing a unified evaluation framework that comprehensively assesses protocols across various metrics, including network throughput, energy consumption, lifetime, latency, and scalability. This holistic approach would provide a more nuanced understanding of protocol performance in diverse network scenarios.

7.2 Real-World Validation and Deployment

Prioritizing field trials and simulations that closely mimic real-world conditions is essential. Collaborations with industry partners or deploying these protocols in experimental testbeds can validate their efficacy, addressing the gap between theoretical proposals and practical implementations.

7.3 Adaptive and Self-Configuring Protocols

Designing adaptive protocols capable of self-configuration and dynamic adjustment to varying network conditions could be pivotal. These protocols would intelligently adapt to changes in network topology, traffic patterns, and energy availability, ensuring robustness and scalability.

7.4 Optimization for Resource-Constrained Devices

Streamlining protocols to reduce computational complexity and resource requirements is crucial, particularly for devices with limited processing power and energy reserves. Creating lightweight versions of these protocols or optimizing algorithms for efficiency could facilitate deployment in IoT devices and low-power networks.

7.5 Standardization and Interoperability

Establishing standardized protocols that ensure interoperability across different network architectures and devices is vital. This would promote seamless integration and adoption, fostering a more cohesive and efficient network infrastructure.

7.6 Security and Privacy Considerations

Integrating robust security mechanisms into these protocols to safeguard against vulnerabilities and ensure user privacy in data transmission is imperative. Future research should emphasize developing protocols with built-in security features to mitigate potential threats. By focusing on these future directions, researchers can address the limitations observed in existing protocols and pave the way for more efficient, adaptable, and practical solutions for wireless network optimization.

Table 2: Comparative analysis of mechanisms for balancing adaptive protocols and energy efficiency in IoT

Ref.	Technology used		Performance metrics				
	AI/ML techniques	Routing protocols	Network throughput	Network lifetime	Energy consumption efficiency	Data packet ratio	Outperforms LEACH, HEED, TEEN, SEP protocols
[22]	Clustering	BMPriority-based CSMA/CA	✓	✓	×	×	✓
[23]	Clustering	Fuzzy C-means	✓	✓	×	×	✓
[24]	Reinforcement learning	IEEE 802.15.4 and the Medium Access Control (MAC)	✓	✓	✓	✓	✓
[25]	Clustering	Advanced Zone-Stable Election Protocol (AZ-SEP).	✓	✓	✓	×	✓
[26]	Clustering	Improved-Adaptive Ranking based Energy-efficient Opportunistic Routing protocol (I-AREOR)	✓	✓	✓	✓	✓
[27]	Deep learning	MAC protocol	✓	✓	×	×	×
[28]	Clustering	particle swarm optimization (PSO) + genetic algorithm (GA) based protocol	✓	✓	✓	✓	✓
[29]	Clustering	IEEE 802.15.4+ an Adaptive Duty cycle (OGMAD)	✓	✓	✓	✓	✓
[30]	Deep learning	heuristic model for multi-stage weighted election heuristic (MSWE) based protocol	✓	✓	✓	✓	✓
[32]	Clustering	MAC protocol	✓	✓	×	×	×
[33]	Clustering	802.15.4 + Kalman filter-based MAC protocol	✓	✓	✓	✓	✓
[34]	Clustering	Multipath Routing Protocol (OQoS-CMRP)	✓	✓	✓	✓	✓
[35]	Clustering	Butterfly Optimization Algorithm (BOA) + Ant Colony Optimization (ACO) based protocol	✓	✓	✓	✓	✓
[36]	Clustering	Fuzzy-GWO based protocol	✓	✓	✓	✓	✓
[37]	Clustering	Grey wolf optimizer (GWO) and the Tabuse arch algorithm (TSA) based protocol	✓	✓	✓	✓	✓
[38]	Clustering	Artificial Bee Colony algorithm-based protocol	✓	✓	✓	✓	✓
[39]	Clustering	Grey wolf optimization algorithm (GWO) based protocol	✓	✓	✓	✓	✓
[40]	Clustering	genetic algorithm (GA) and particle swarm optimization-based protocol	✓	✓	✓	✓	✓

8. Conclusion

This paper summarizes the primary studies related to Energy Consumption and Connectivity through energy-efficient Protocols for Sustainable Networking in Smart Cities. The comparative analysis of ML-based systems shows that efficient protocols are better than conventional protocols to provide a secure and efficient communication path between the sensor nodes in wireless sensor networks. Moreover, the integration of these protocols with AI/ML-based mechanisms enhances the network throughput, lifetime, energy efficiency, and latency. These findings offer promising approaches to reduce energy dissipation in IoT-based smart cities and address the pressing need to mitigate carbon footprints associated with the rapid proliferation of sensor devices. Controlling these carbon footprints while developing smart cities is a crucial requirement that needs to be tackled with the utmost priority.

Efficient energy protocols offer enhanced performance metrics and contribute to the overall sustainability and resilience of smart city infrastructures. By optimizing energy consumption and improving network efficiency, these protocols play a crucial role in mitigating environmental impact while ensuring reliable connectivity and service delivery. Despite advancements in this field, there is still huge potential for improvement. This paper contributes in the same direction by providing a review of energy-efficient protocols to achieve maintainability in smart cities.

In conclusion, the review of energy-efficient protocols presented in this paper serves as a valuable resource for researchers, policy-makers, and practitioners striving toward the goal of sustainable networking in smart cities. For researchers, this review offers a comprehensive overview of the latest advancements in energy-efficient protocols for smart city networking. By synthesizing existing literature and analyzing various technologies and techniques, researchers gain insights into the state-of-the-art approaches and emerging trends in the field. This serves as a foundation for further research and innovation, guiding future studies aimed at optimizing energy consumption, enhancing network performance, and reducing carbon footprints in smart city infrastructures.

Policy-makers can leverage the findings of this review to inform the development of regulations, standards, and incentives aimed at promoting energy efficiency and sustainability in smart cities. By understanding the efficacy of different energy-efficient protocols and their potential impact on carbon emissions and environmental sustainability, policymakers can formulate evidence-based policies to support the transition towards greener and more resilient urban environments. Practitioners involved in the design, implementation, and management of smart city infrastructures can benefit from this review by gaining insights into best practices and practical considerations for deploying energy-efficient protocols. By adopting these protocols and integrating AI/ML techniques, practitioners can optimize network performance, enhance energy efficiency, and reduce operational costs while maintaining reliable connectivity and service delivery for smart city residents and businesses.

To summarize it all, the review of energy-efficient protocols serves as a valuable resource for advancing the sustainability agenda in smart cities. By fostering collaboration between researchers, policy-makers, and practitioners, this review facilitates the development and implementation of innovative solutions to address the challenges of energy consumption, carbon emissions, and environmental impact in smart city networking.

References

[1] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of things is the backbone," IEEE Consumer Electronics Magazine, vol. 5, no. 3, pp. 60-70, 2016

[2] M. A. Ahad, S. Paiva, G. Tripathi, and N. Feroz, "Enabling technologies and sustainable smart cities," Sustainable Cities and Society, vol. 61, p. 102301, 2020.

[3] S. H. Alsamhi, O. Ma, M. S. Ansari, and Q. Meng, "Greening Internet of Things for greener and smarter cities: a survey and future prospects," Telecommunication Systems, vol. 72, pp. 609-632, 2019.

[4] D. Bruneo, S. Distefano, M. Giacobbe, A. L. Minnolo, F. Longo, G. Merlino, and N. Tapas, "An IoT service ecosystem for smart cities: The# smartme project," Internet of Things, vol. 5, pp. 12-33, 2019

[5] Y. T. U. Hasmawaty and D. Antoni, "Building green smart city capabilities in South Sumatra, Indonesia," Sustainability, vol. 14, no. 13, pp. 7695, 2022

[6] M. A. Albreem, A. M. Sheikh, M. J. Bashir, and A. A. El-Saleh, "Towards green Internet of Things (IoT) for a sustainable future in Gulf Cooperation Council countries: Current practices, challenges and future prospective," Wireless Networks, vol. 29, no. 2, pp. 539-567, 2023.

[7] P. Pattnayak, O. P. Jena, and S. Sinha, "Cloud and Green IoT-based technology for sustainable smart cities," Green Engineering and Technology, pp. 1-19, 2021.

[8] G. Maccani, N. Connolly, S. McLoughlin, A. Puvvala, H. Karimikia, and B. Donnellan, "An emerging typology of IT governance structural mechanisms in smart cities," Government Information Quarterly, vol. 37, no. 4, p. 101499, 2020.

[9] Q. V. Khanh, V. H. Nguyen, Q. N. Minh, A. D. Van, N. Le Anh, and A. Chehri, "An efficient edge computing management mechanism for sustainable smart cities," Sustainable Computing: Informatics and Systems, vol. 38, p. 100867, 2023.

[10] P. Chithaluru, F. Al-Turjman, M. Kumar, and T. Stephan, "Energy-balanced neuro-fuzzy dynamic clustering scheme for green & sustainable IoT based smart cities," Sustainable Cities and Society, vol. 90, p. 104366, 2023.

[11] P. He, N. Almasifar, A. Mehbodniya, D. Javaheri, and J. L. Webber, "Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review," Sustainable Computing: Informatics and Systems, vol. 36, p. 100822, 2022.

[12] M. A. Ahad, S. Paiva, G. Tripathi, and N. Feroz, "Enabling technologies and sustainable smart cities," Sustainable Cities and Society, vol. 61, pp. 102301, 2020.

[13] M. Angelidou, A. Psaltoglou, N. Komninos, C. Kakderi, P. Tsarchopoulos, and A. Panori, "Enhancing sustainable urban development through smart city applications," Journal of Science and Technology Policy Management, vol. 9, no. 2, pp. 146-169, 2018.

[14] A. E. Akin-Ponnle and N. B. Carvalho, "Energy harvesting mechanisms in a smart city—A review," Smart Cities, vol. 4, no. 2, pp. 476-498, 2021.

[15] G. Verma and S. Prakash, "A comparative study based on different energy saving mechanisms based on green Internet of Things (GIoT)," in 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 659-666, June 2020.

[16] S. H. Alsamhi, O. Ma, M. S. Ansari, and Q. Meng, "Greening Internet of Things for greener and smarter cities: a survey and future prospects," Telecommunication Systems, vol. 72, pp. 609-632, 2019

[17] H. S. Dunn, "The carbon footprint of ICTS," Global Inf. Soc. Watch, University of West Indies, Kingston, Jamaica, 2010.

[18] M. A. Albreem, A. M. Sheikh, M. H. Alsharif, M. Jusoh, and M. N. M. Yasin, "Green Internet of Things (GIoT): Applications, practices, awareness, and challenges," IEEE Access, vol. 9, pp. 38833-38858, 2021

[19] Z. Y. M. Yusoff, M. K. Ishak, and K. A. Alezabi, "The role of RFID in green IoT: A survey on technologies, challenges and a way forward," Adv. Sci. Technol. Eng. Syst. J, vol. 6, no. 1, pp. 17-35, 2021.

[20] C. Baldé, V. Forti, V. Gray, R. Kuehr, and P. Stegmann, "The Global E-Waste Monitor–2017," United Nations University (UNU), International Telecommunication Union (ITU) & International Solid Waste Association (ISWA), Bonn/Geneva/Vienna, Standard, Electronic Version, pp. 978–992, 2017

[21] Guardian, "Tsunami of Data' Could Consume One Fifth of Global Electricity by 2025," Accessed: May 5, 2020. [Online]. Available: https://www.theguardian.com/environment/2017/dec/11/tsunami-ofdata-could-consume-fifth-global-electricity-by-2025/

[22] I. Dbibih, I. Iala, F. Z. E. Biach, and O. Zytoune, "An efficient algorithm for end-to-end latency optimization over IEEE 802.15.4 wireless networks for IoT applications," International Journal of Intelligent Engineering & Systems, vol. 14, no. 5, pp. 336-347, 2021

[23] N. Rani and P. Kumar, "Energy efficient hierarchical routing protocols for IoT," Int. J. Eng. Adv. Technol. (IJEAT), vol. 8, no. 6, 2019

[24] N. Charef, "Energy Sustainable Reinforcement Learning-based Adaptive Duty-Cycling in Wireless Sensor Networks-based Internet of Things Networks," 2023

[25] A. K. Rana and S. Sharma, "Enhanced energy-efficient heterogeneous routing protocols in WSNs for IoT application," IJEAT, vol. 9, no. 1, pp. 4418-4415, 2019

[26] P. Chithaluru, F. Al-Turjman, M. Kumar, and T. Stephan, "I-AREOR: An energy-balanced clustering protocol for implementing green IoT in smart cities," Sustainable Cities and Society, vol. 61, p. 102254, 2020

[27] K. Haseeb, I. U. Din, A. Almogren, I. Ahmed, and M. Guizani, "Intelligent and secure edge-enabled computing model for sustainable cities using green internet of things," Sustainable Cities and Society, vol. 68, p. 102779, 2021.

[28] M. Mishra, G. S. Gupta, and X. Gui, "Network Lifetime Improvement through Energy-Efficient Hybrid Routing Protocol for IoT Applications," Sensors, vol. 21, no. 22, p. 7439, 2021.

[29] A. N. Alvi, S. Khan, M. A. Javed, K. Konstantin, A. O. Almagrabi, A. K. Bashir, and R. Nawaz, "OGMAD: Optimal GTS-allocation mechanism for adaptive data requirements in IEEE 802.15.4 based Internet of Things," IEEE Access, vol. 7, pp. 170629-170639, 2019.

[30] A. S. H. Abdul-Qawy and T. Srinivasulu, "SEES: A scalable and energy-efficient scheme for green IoT-based heterogeneous wireless nodes," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 4, pp. 1571-1596, 2019.

[31] A. H. Sodhro, S. Pirbhulal, Z. Luo, and V. H. C. De Albuquerque, "Towards an optimal resource management for IoT based Green and sustainable smart cities," Journal of Cleaner Production, vol. 220, pp. 1167-1179, 2019.

[32] M. A. Alharbi, M. Kolberg, and M. Zeeshan, "Towards improved clustering and routing protocol for wireless sensor networks," EURASIP Journal on Wireless Communications and Networking, vol. 2021, no. 1, pp. 1-31, 2021.

[33] Iala, I. Dbibih, O. Zytoune, M. Rziza, and D. Aboutajdine, "A Kalman Filter Process for Energy Optimization in WSNs," Journal of Communications Software and Systems, vol. 15, no. 1, pp. 9-17, 2019.

[34] O. Deepa and J. Suguna, "An optimized QoS-based clustering with multipath routing protocol for wireless sensor networks," Journal of King Saud University-Computer and Information Sciences, vol. 32, no. 7, pp. 763-774, 2020.

[35] P. Maheshwari, A. K. Sharma, and K. Verma, "Energy-efficient cluster-based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization," Ad Hoc Networks, vol. 110, p. 102317, 2021.

[36] M. Rami Reddy, M. L. Ravi Chandra, P. Venkatramana, and R. Dilli, "Energy-efficient cluster head selection in wireless sensor networks using an improved grey wolf optimization algorithm," Computers, vol. 12, no. 2, p. 35, 2023.

[37] R. A. Sadek, D. M. Abd-alazeem, and M. M. Abbassy, "A new energy-efficient multi-hop routing protocol for heterogeneous wireless sensor networks," International Journal of Advanced Computer Science and Applications, vol. 12, no. 11, 2021

[38] Z. Wang, H. Ding, B. Li, L. Bao, and Z. Yang, "An energy-efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks," IEEE Access, vol. 8, pp. 133577-133596, 2020.

[39] M. Elhoseny, R. S. Rajan, M. Hammoudeh, K. Shankar, and O. Aldabbas, "Swarm intelligence–based energy efficient clustering with multihop routing protocol for sustainable wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 16, no. 9, pp. 1550147720949133, 2020.

[40] B. M. Sahoo, H. M. Pandey, and T. Amgoth, "GAPSO-H: A hybrid approach towards optimizing the cluster-based routing in wireless sensor network," Swarm and Evolutionary Computation, vol. 60, p. 100772, 2021.

[41] City of Copenhagen. "Copenhagen City of Solutions." Retrieved from https://stateofgreen.com/en/partners/city-of- copenhagen/-

The Nucleus ***61***, *No. 1 (2024) 51-62*

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

A Comprehensive Review of Machine Learning-Based Malware Detection Techniques for Windows Platform

Amna Wajid[1*], Tauqir Ahmed[2], Uroosa Bilal Chaudhry[1]

[1]*Department of Computer Science, Rachna College of Engineering and Technology, Gujranwala, Pakistan*

[2]*Department of Computer Science, University of Engineering and Technology, Punjab, Pakistan*

ABSTRACT

The growing threat of windows malware poses an increasing risk to the security of computers and the sensitive information they hold. The exponential rise in malware threats targeting the Windows platform necessitates robust and adaptive detection mechanisms. Machine learning (ML) techniques demonstrate effectiveness in identifying windows malware, a thorough analysis of these techniques is essential. This paper presents a comprehensive review of machine learning based techniques which have been proposed by research community for detecting windows malware. The review begins by providing a comparison of this study with the existing reviews. Then, we provide details of different ML based malware detection techniques. These techniques have been assessed on multiple parameters including dataset used for training and testing, availability of dataset, ML model used for classification, the type of extracted features, analysis type and the metrics employed to measure the effectiveness of technique. Furthermore, the paper highlights the limitations and challenges in this field and suggests potential future research directions. By providing a comprehensive overview and critical analysis of ML-based malware detection techniques proposed for the Windows environment, this study aims to guide and inspire further research in handling evolving cyber threats.

Keywords: *Malware Detection, Windows Platform, Machine Learning*

1. Introduction

Now a days, there is a significant surge in the adoption of Windows operating systems. The appeal of Windows OS lies in its user-friendly interface, facilitating effortless computer operations for individuals with varied technical expertise. Its widespread popularity is attributed to a multitude of features, including hardware independence, compatibility with third-party software, open-source options, entertainment functionalities, and various other capabilities. However, the extensive use of Windows OS has also attracted malicious actors seeking to exploit its popularity. This heightened interest from attackers has led to the propagation of novel malware types, posing a threat to the confidentiality and integrity of data stored on these systems [1].

Over the past decade, there has been a significant proliferation in the development of computer malware. In the current landscape, cybercriminals use malware as a potent weapon to execute attacks on computer systems. The internet serves as the primary medium for launching malware attacks, utilizing channels such as emails, malicious websites and the distribution of software through drives and downloads. Malicious software encompasses a diverse array, including viruses, Trojan horses, worms, root kits, adware or ransomware [2].

To discern between malware and benign samples, analysts employ static or dynamic analysis techniques. Static analysis, also known as signature-based analysis, identifies malicious behaviors within binary source code segments without executing the applications. On the other hand, dynamic analysis detects malware during the execution of malicious applications by observing their characteristics while running on a host system. Typically conducted in a sandbox Environment-a controlled, isolated environment where untrusted software can be run securely and separately from the rest of the system without damaging it, this analysis prevents the actual infection of production systems by malware. However, it is important to note that dynamic analysis consumes more resources and entails higher costs [1].

With the increasing frequency of malware attacks, the significance of deploying reliable classification and detection methods increases. The established realm of computer science, namely machine learning, exhibits significant potential in the realm of windows malware detection. With the capability to discern complex data patterns and acquire knowledge from diverse datasets, algorithms of machine learning prove to be well-suited for the identification of computer malware. The increasing interest in employing ML techniques for malware detection has led to a proliferation of studies in this domain. However, owing to the dispersed nature of existing research in this field, there arises a need for a comprehensive review of ML based techniques for windows malware detection [3].

There exist a lot of comprehensive and systematic review papers for android based malware detection techniques. However, despite the abundance of literature in this area, there remains a notable scarcity in reviews addressing malware detection for the Windows platform. Moreover, the existing reviews are predominantly informal and exhibit a limited scope. So, to address this gap, this study aims to present a thorough examination of the current state of the research in windows malware detection through machine learning. The review systematically delves into various machine-learning techniques employed for detecting malware, scrutinizes the metrics which are used for evaluation of performance and presents the limitations and challenges of the currently

*Corresponding author: engramnawajid@gmail.com

employed methods. The final section of our review identifies potential avenues for future research in this domain.

2. Literature Review

As the usage of Windows OS has become widespread, the corresponding increase in security threats from malware has fueled substantial advancements in machine learning-based malware detection in recent times [3]. For providing researchers a review of existing techniques, a lot of review studies exist covering the malware detection techniques for android as well as windows platform.

Ö. A. Aslan and R. Samet have performed a survey on various malware detection techniques. The techniques covered in this survey are heuristic-based, signature-based, behavior-based, cloud-based, model checking-based, mobile devices-based, DL-based and IoT-based techniques. In this review, a summary of malware detection techniques has been presented. Also, the current challenges associated with malware detection and the future recommendations to overcome challenges have been given in this review. The limitation of this review is that it has covered studies till 2019. Also, the survey approach is informal and do not give an in depth review of existing approaches for malware detection [4].

Q. Wu, X. Zhu and B. Liu have performed a review on ML based methodologies for static android malware detection. In the review, authors have delved into the structural aspects of Android applications, examined a range of static feature resources, assessed machine learning techniques for identifying Android malware, discussed both the merits and constraints associated with these approaches and given future directions for researchers who want to work in this domain. This survey has covered studies up to 2020 and also it has only covered static features based techniques. So, this review could be improved by incorporating dynamic feature extraction techniques and by including recent papers from 2021 onwards [6]

E. J. Alqahtani *et. al.* have surveyed ML based malware detection methodologies for android devices. The primary objective of the paper is to survey and summarize the different methods and approaches for detecting Android malware, particularly those that leverage machine learning algorithms. The main limitation of this paper is that, it does not provide in-depth analysis or evaluation of each individual technique. Readers seeking a more detailed understanding of specific methods may need to refer to the original research papers. Moreover, the paper was published in 2019, which means it might lack the incorporation of the latest advancements and techniques in detecting Android malware. The field evolves rapidly, and new approaches or trends may have emerged since the paper's publication [7].

M. N. U. R. Chowdhury *et. al.* have reviewed the current research trends on malware detection techniques on android platform using ML. Authors have provided an overview of Android malware and the resulting security concerns caused by it. Then, they presented various unsupervised, supervised, and DL methodologies that have been used for detection of android malware. But, the data is not presented in tabular form so, it is difficult to analyze the data given in this paper. Moreover, the paper covers malware detection techniques for android platform only [3].

V. Kouliaridis and G. Kambourakis have conducted a comprehensive survey of cutting-edge techniques for detecting Android malware using machine learning. They achieved this by categorizing and providing concise analyses of the latest research over the past seven years, from 2014 to 2021. Their categorization involved examining the analysis type, feature extraction methods, datasets, machine learning classification techniques and the performance evaluation metrics used in these studies. Furthermore, they offered detailed insights into their findings, emerging research trends, potential challenges and future research directions. It's crucial to note that this survey focuses exclusively on ML techniques for Android malware detection and encompasses research up to the year 2021 [8].

J. Senanayake *et. al.* have conducted a review of the literature with the aim of focusing on the use of ML for malware detection on android mobile devices. The paper categorizes the different machine learning techniques and their effectiveness. The authors have discussed the criteria utilized in assessing the effectiveness of different approaches to detect malware such as precision, accuracy, recall and F1-measure. Authors have also identified common challenges in Android mobile malware detection and highlighted emerging trends in the field, shedding light on potential areas for future research. The paper provides insights up to 2021. The field of android malware detection and ML continues to evolve rapidly. Therefore, the review does not include the most recent advancements and techniques developed after 2021. So, this work could be extended further by including studies after 2021 [9].

M. Al-Janabi and A. M. Altamimi have presented a review which covers the malware detection methods using ML along with basic concepts of both topics of malware detection and ML. Various representative research studies were reviewed and classified according to their analysis techniques, whether they employed static, dynamic or hybrid approaches. But the scope of paper is very limited. It has given review of only 10 techniques and covers studies published up to 2019 [10].

In literature, several review paper exist for android malware detection. In [11, 12, 13], review of ML based android malware detection approaches have been presented. But we have found only few reviews on windows based malware detection techniques and those existing reviews have a limited scope and cover a few ML based techniques and those reviews are published in 2020 so, they do not cover work done after 2020. A comparison of existing surveys has been performed based on following factors and has been shown in Table 1.

Table1: The Comparison of Existing Surveys

Reference	Year	Survey Approach	Malware Detection Approaches Reviewed		Feature Extraction Type		Malware Detection for Platform		Papers Included till
			Machine Learning based	Traditional Approaches	Static	Dynamic	Android	PC	
[4]	2020	Informal	✗	✓	✗	✗	✗	✓	2019
[6]	2021	Formal	✓	✗	✓	✗	✓	✗	2020
[7]	2019	Informal	✓	✗	✓	✗	✓	✗	2018
[3]	2023	Systematic	✓	✗	✓	✓	✓	✗	2022
[8]	2021	Formal	✓	✗	✓	✓	✓	✗	2021
[9]	2021	Systematic	✓	✗	✓	✓	✓	✗	2021
[10]	2020	Formal	✓	✗	✓	✓	✗	✓	2019
This Study	2023	Formal	✓	✗	✓	✓	✗	✓	2023

1. *Reference:* This column indicates the reference number of each study
2. *Year:* This column represents the year in which the study was conducted or published.
3. *Survey Approach:* This column specifies the approach used in conducting the survey or review. It can be informal, or systematic.
4. *Malware Detection Approaches Reviewed:* This column indicates whether the study reviewed machine learning-based approaches (✓) or traditional approaches (✗) for malware detection.
5. *Feature Extraction Type:* This column specifies the type of feature extraction techniques used in the study for malware detection.
6. *Malware Detection for Platform:* This column indicates whether the study focused on Android (✓) or PC (✓) platforms for malware detection.
7. *Papers Included till:* This column represents the time span of papers included in the selected study.

3. Methodology

3.1 Selection Criteria

A set of criteria has been established to select the studies to incorporate in this survey. These criteria aim to offer a thorough examination of the diverse ML approaches employed for detecting Windows malware. Furthermore, the selected criteria will provide a thorough understanding of the present research landscape and its applicability to malware detection. The selection criteria encompass:

1. Relevance: This involves research focused on malware detection on windows OS using ML algorithms.
2. Year of Publication: To stay updated about the latest developments, studies published between 2019 and 2023 have been selected to include in this study.
3. Data Availability: Those studies have been included in this review that have made their dataset accessible to the public research community or have provided sufficient information to enable result reproduction.
4. Methodology: Studies included in this survey, particularly those related to windows malware detection, must employ ML algorithms.
5. Evaluation: The evaluations of ML algorithms in this survey rely on quantitative metrics such as accuracy and robustness.

3.2 Selection of the Studies

An extensive search was conducted across various platforms, including online databases such as IEEE Xplore, Springer, ACM Digital Library, Science Direct and Google Scholar to identify pertinent studies for this study. A collection of keywords associated with windows malware detection and ML guided the search. To search the relevant papers, the following search query was used;

(Artificial Intelligence OR Machine Learning OR Deep Learning) AND (Malware OR Malicious Code OR Virus OR Worms) AND (Detection OR Identification OR Recognition) AND (Technique* OR Method OR Approach)

The preliminary search produced a multiple results, which were subsequently refined based on the selection criteria mentioned in the previous section. To assess the relevance and appropriateness of each study for inclusion in this survey, a meticulous examination of both the abstract and full text was undertaken during this filtering process.

3.3 Data Synthesis and Analysis

The chosen papers underwent a comprehensive scrutiny in

Ref.	Year	ML Models user for Classifications	Dataset	Dataset Availability	Deep Learning Model used	Feature Extracted from	Best Model	Can detect unknown malware	Analysis Tpye

Fig. 1. Data extracted from different studies

the course of data collection and analysis to acquire relevant insights into windows malware detection utilizing machine learning. To guarantee the consistency, comprehensiveness and currency of the findings in this review, the information was systematically gathered. The details extracted from each paper are as shown in Fig. 1

Moreover, the information derived from the chosen papers served the purpose of comparing different approaches and identifying potential avenues for future investigation. Through this analysis, a thorough comprehension of the current state of the field was presented, along with highlighting the primary challenges and opportunities for future research.

4. Overview of Selected Studies and Key Finding

S. Naz, and D. K. Singh have introduced a system employing a static analysis approach to preemptively detect malware before the installation of executable files. Their methodology comprises four primary steps: (1) collecting samples, (2) extracting features, (3) dividing the dataset and (4) classifying executable files. The malware analysis involves comparing the extracted features of the PE file. Feature extraction relies on recent research, leveraging prior knowledge of the PE file header. Various techniques like n-datagram, grayscale, and others are utilized for this purpose. In their proposed model, the authors consider the entirety of the PE file header to evaluate features from the executable files. These features are subsequently employed to construct a classifier, aiding in determining the file's malicious nature [1].

A. Hussain *et. al.* have presented a machine learning-driven method for identifying malicious software in the Windows operating system. The proposed model examines distinct characteristics of the PE header file, which it then contrasts with a trained machine learning model to determine if the file is malicious or benign. The study employs six distinct machine learning algorithms to identify malware in Windows executable files. The performance of these algorithms is evaluated based on metrics like precision, recall, accuracy and F1-score. The experimental findings indicate that RF outperforms other algorithms, achieving an impressive accuracy of 99.4% [14].

R. Damaševičius *et. al*. have introduced a methodology for malware detection based on ensemble classification. It uses a two-step classification method, starting with a stacked ensemble of dense and CNN in the first stage, and then employing a meta-learner for the final stage classification. The study explores and compares 14 different classifiers for the meta-learner. For comparison, 13 machine learning techniques are utilized, including kNN, Linear SVM, Radial basis function (RBF) SVM, RF, AdaBoost, DT, ExtraTrees, Linear Discriminant Analysis, , Passive Classifier, Logistic Neural Net , Stochastic Gradient Descent classifier and Ridge Classifier. The experiments have been performed using PE headers (ClaMP) dataset. The optimal performance is achieved with an ensemble consisting of five dense and CNN neural networks, combined with the ExtraTrees classifier as the meta-learner [15].

M. Almousa *et. al.* have emphasized an approach to ransomware detection using Application Programming Interfaces (APIs) in conjunction with ML techniques. The primary objectives of this study include (i) gaining insights into the ransomware lifecycle on windows platform, (ii) conducting dynamic analyses of samples of ransomware to capture several features associated with malicious code patterns and (iii) the development and validation of ransomware detection models based on machine learning, utilizing diverse benign and ransomware samples. Data was gathered from publicly available repositories and underwent analysis using sandbox for sampling purposes. The acquired datasets were employed for constructing machine learning models. The analysis resulted in an impressive ransomware detection accuracy of 99.18% on windows platforms, demonstrating the potential for achieving highly accurate ransomware detection by combining API calls and machine learning models [16].

A. Irshad *et. al.* have proposed a ML based technique for malware detection. In their work, authors have extracted features from JSON reports generated by Cuckoo for Windows executable files. These features consist of sensitive and confidential data repeatedly found in the JSON report. After feature extraction, a Genetic Algorithm has been used to identify the most significant optimal features. These selected optimal features are then used as input to train classifiers capable of distinguishing between Malware and Benign files. SVM achieved an accuracy of 81.3%, the NB classifier reached an accuracy of 64.7%, and the RF classifier demonstrated an accuracy of 86.8% [17].

F. O. Catak *et. al.* have presented their work with primary objective of creating a classification approach for distinguishing various malware types based on their behavior. The research began by constructing a novel dataset that captures the API calls made within the windows OS, reflecting the behavior of malicious software. The dataset encompasses a range of malicious malware types, including Backdoor, Adware, Downloader, Spyware, Dropper, Trojan horse, worm and virus. For classification task, the LSTM (Long Short-Term Memory) method, a well-established technique for handling sequential data has been employed. The outcomes achieved by classifier exhibit an impressive accuracy of up to 95% and an F1-score of 0.83, which is highly satisfactory [18].

X. Huang *et. al.* have introduced a novel approach for malware detection that leverages deep learning techniques. This method integrates malware visualization technology with a CNN and the neural network architecture is based on the VGG16 network. Authors have put forth a hybrid visualization technique for malware which combines insights from both static and dynamic analyses [19].

D. Rabadi and S. G. Teo have explored a novel approach to extract dynamic features based on API calls by examining

both the API calls and their corresponding list of arguments. By harnessing machine learning algorithms, Authors have developed two methods to identify and categorize Windows malware samples. The first method treats the entire list of arguments of each API call as a single feature, while the second method treats each argument of each API call individually as a feature. The findings demonstrate that the proposed approach surpasses recent malware detection methods based on API arguments in terms of accuracy, constraints and the amount of API-related information required. Through experiments, a remarkable accuracy rate of over 99.8992% has been achieved, which outperforms the current state-of-the-art approaches [20].

K. Sethi *et. al.* have introduced an innovative framework for malware analysis, designed to efficiently detect and categorize malware. The proposed approach hinges on the utilization of two distinct feature selection algorithms to extract the most pertinent features. This not only reduces training time but also improves the accuracy of classification and detection. The experimental findings highlight that the Decision Tree classifier, in particular, yields a high level of accuracy when compared to other classification methods [21].

Ö. Aslan and A. A. Yilmaz have introduced an innovative deep learning architecture designed for the classification of malware variants through a hybrid model. The primary contribution made by this research lies in the introduction of a new hybrid architecture that optimally combines two extensive pre-trained network models. This architecture is structured into four key stages: data gathering, the design of a deep neural network (DNN) framework, the training of the proposed DNN and evaluation of the trained DNN. The results demonstrate that the proposed approach can classify malware with a notably high level of accuracy, surpassing the performance of existing methods in the field. When evaluated on the Malimg dataset, it achieved an accuracy of 97.78%, outperforming the majority of machine learning-based malware detection techniques [5].

M. S. Akhtar and T. Feng have introduced a protective mechanism that assessed three machine learning algorithm approaches for malware detection, ultimately selecting the most suitable one. The findings revealed that, in terms of detection accuracy, Decision Trees (DT) performed exceptionally well at 99%, followed by CNN at 98.76% and SVM at 96.41% when compared to other classifiers. The authors systematically evaluated and quantified the detection accuracy of a machine learning classifier employing static analysis to extract features based on PE data, contrasting its performance with two alternative machine learning classifiers [22].

K. Shaukat *et. al.* have proposed a novel approach which introduces a hybrid framework that merges deep transfer learning and ML for malware detection. Initially, deep transfer learning is employed to extract comprehensive deep features from the last fully connected layer of the deep learning model. Subsequently, machine learning models serve as the final detector, effectively leveraging the intrinsic connections between input and output. The efficacy of the suggested framework is confirmed through validation on a compact dataset. The performance of different models has been assessed, initially focusing on a single feature and subsequently incorporating all features for malware classification. The findings indicate that the proposed framework outperforms other contemporary techniques in terms of effectiveness [23].

M. S. Akhtar and T. Feng have constructed an innovative ensemble of deep neural networks by combining CNN and LSTM techniques. The CNN-LSTM method we introduced is specifically designed for advanced malware detection without the need for feature engineering. The proposed CNN-LSTM approach achieves the highest detection accuracy, reaching 99%, surpassing other methods for malware detection [24].

M. Ahmed *et. al.* have represented malware signatures as 2D images and employ deep learning techniques to characterize these signatures within the BIG15 dataset covering nine classes. The proposed work assesses the performance of diverse ML and DL technologies for malware classification, including Logistic Regression (LR), CNN, Artificial Neural Network (ANN), transfer learning with CNN and LSTM. The transfer learning technique, specifically utilizing InceptionV3, demonstrates notable performance, surpassing models such as LSTM, achieving an accuracy of 98.76% on the testing dataset and 99.6% on the training dataset [25].

J. Palša *et. al.* have concentrated on training ML models employing XGBoost and extremely randomized trees algorithms on two datasets derived from static and dynamic analyses of authentic benign and malicious samples. Subsequently, a comparative analysis of their success rates was conducted, both against each other and in comparison to additional algorithms, including RF, DT, SVM and naïve Bayes previously assessed in authors prior study on the similar datasets. The most effective classification models, utilizing XGBoost algorithm, demonstrate remarkable performance metrics, achieving a detection accuracy of 91.9% on the static analysis dataset. Similarly, on the dynamic analysis dataset, the XGBoost models achieve a detection accuracy of 96.4% [26].

The Study by G. O. Ganfure *et. al.* has been driven by the necessity for improved ransomware detection techniques capable of identifying both known and new ransomware types effectively and efficiently. This research introduces "DeepWare," a model to detect ransomware that merges DL with hardware performance counter (HPC) insights. Unlike previous approaches that analyze all HPC data at a single time point for each process, DeepWare adopts a more streamlined strategy visualizing HPC data using deep learning to efficiently and effectively identify ransomware. Experimental results across various ransomware types show DeepWare achieving a 98.6% recall score, outperforming existing similar approaches like RATAFIA, OC-SVM and EGB models by 84.41%, 60.93%, and 21%, respectively [27].

U. Zahoora *et. al.* have introduced CSPE-R, a Cost-Sensitive Pareto Ensemble strategy designed for detecting new ransomware threats. Initially, the framework utilizes an unsupervised deep Contractive Auto Encoder (CAE) to transform the complex feature space into a more uniform and fundamental semantic space. To develop robust features, CSPE-R explores various semantic spaces at different levels of intricacy. Diverse base estimators are trained across these derived subspaces to establish critical connections among different ransomware attack families. Subsequently, a unique Pareto Ensemble-based approach is employed to select the most effective estimators, achieving a balance between false positives and false negatives. Ultimately, the decisions from these chosen estimators are combined to enhance detection capabilities against unfamiliar ransomware threats. Experimental findings demonstrate CSPE-R's effectiveness in identifying zero-day ransomware attacks [28].

P. Tumuluru *et. al.* have employed machine learning-based static malware research systems to identify Windows-based malwares. The proposed methodology comprises several distinct phases. Firstly, it involves gathering a comprehensive dataset encompassing both malware and non-malware files. Subsequently, a curated dataset is created, followed by the generation of a detailed report. The process entails extracting a myriad of characteristics from the data, encompassing diverse features. Employing a Genetic Algorithm, the methodology undergoes a meticulous feature selection phase. Finally, detection is executed using different machine learning classifiers including KNN, RF, LR, XGBoost to identify and classify malware instances effectively. After analyzing all the results, it was found that RF Classifier and XGBoost Classification works gives more accuracy as compared to KNN and LR [29].

M. Kumar has introduced a scalable malware detection system utilizing big data and a machine learning framework. The machine learning model, implemented via Apache Spark to facilitate distributed learning, employs locality-sensitive hashing for efficient malware detection, notably decreasing detection time. The implementation and experimental analysis follow a five-stage iterative process. The model proposed in this study demonstrates a remarkable 99.8% accuracy rate. Moreover, compared to models proposed by other researchers, this approach significantly diminishes learning and malware detection duration [30].

S. S. Alshamrani has proposed a novel PDF malware identification system utilizing ML. The uniqueness of this system lies in its dual inspection of PDF files statistically and dynamically resulting in heightened accuracy in identifying the document's nature. Operating without signatures, this method holds promise in discerning unfamiliar and zero-day malware. The experiment evaluates five distinct classifier algorithms to determine the most suitable fit. Assessment metrics such as true positive rate (TPR), precision, false positive rate (FPR), false negative rate (FNR) and F1-score are computed for each classifier algorithm to identify the best approach. Comparative analysis is conducted against existing PDF classification systems. Additionally, a malicious attack simulation is executed, concealing the malicious code within the PDF file during parsing by the PDF parser. The proposed technique achieves an F1-score of 0.986 using the RF classifier, surpassing the cutting edge F1-measure of 0.978. Thus, this method exhibits effectiveness in detecting malware embedded within PDF files compared to existing systems [31].

F. Alhaidari *et. al.* has proposed a system, named Zero-Day Vigilante (ZeVigilante), with aim to identify malware by integrating both static and dynamic analyses. In contrast to previous studies, the proposed approach incorporates substantial datasets encompassing ample samples for both types of analyses. These meticulously processed datasets serve as the foundation for training and testing various ML classifiers, including RF, NN, DT, kNN, NB and SVM. Notably, the Random Forest (RF) classifier achieves the highest accuracy rates, recording 98.21% for static analysis and 98.92% for dynamic analysis [32].

W. Z. Zakaria *et. al.* have introduced RENTAKA, a machine learning framework specifically designed for the early detection of crypto-ransomware. Extracted features align with different phases within the ransomware lifecycle. The experimental phase involved assessing five commonly used machine learning classifiers: Naïve Bayes, kNN, Support Vector Machines, Random Forest and J48. This research presents a pre-encryption detection framework for crypto-ransomware utilizing a machine learning approach. Results indicate that support vector machines (SVM) exhibited the highest accuracy and TPR, achieving 97.05% accuracy and a TPR of 0.995 based on experiments [33].

E. V. P. Kalyan *et. al.* have presented a malware detection system based on DL with focus on detection and categorization of harmful software. This study has introduced a highly accurate and efficient malware detection method utilizing convolutional neural networks (CNNs). The system takes binary files as input and distinguishes between harmful and benign ones. Minimal preprocessing is applied to the binaries, and the network is responsible for discovering features during training an important deviation from current convolutional neural networks. The CNN algorithm demonstrates higher accuracy and efficiency compared to alternative algorithms. Upon implementing the algorithm, authors have attained a commendable accuracy of 95% [34].

Q. Abu Al-Haija *et. al.* have presented a novel detection system designed to analyze PDF documents and differentiate between benign and malware-infected PDF files. The system proposed here utilizes the AdaBoost decision tree, optimized with ideal hyper parameters, trained, and assessed on an extensive and contemporary dataset named Evasive-PDFMal2022. The experimental evaluation showcases an efficient PDF detection system, having an impressive 98.84% accuracy within a brief prediction interval of 2.174 seconds. Consequently, this model surpasses other cutting-edge models within the same research domain. Thus, the proposed system proves to be an effective tool for identifying PDF

malware, exhibiting high detection performance while minimizing detection overhead [35].

N. A. Azeez *et. al.* have proposed an ensemble learning approach for malware detection, employing a stacked ensemble of fully-connected networks and one-dimensional CNNs for initial classification. Subsequently, a machine learning algorithm has been utilized for final-stage classification. In the selection of a meta-learner, authors have scrutinized and compared 15 ML classifiers. Additionally, for comparative purposes, five machine learning algorithms NB, DT, RF, AdaBoosting and gradient boosting were employed. The experiments were conducted on the windows PE malware dataset. The most promising outcomes were achieved using an ensemble comprising seven NN, along with the ExtraTrees classifier serving as the last-stage classifier [36].

N. Loi *et. al.* have introduced a malware classification pipeline designed to categorize Windows Portable Executable files (PEs). Upon receiving a PE sample, the pipeline initially determines whether it's malicious or benign. If classified as malicious, the pipeline proceeds to conduct a detailed analysis to identify its threat type, family, and behavioral traits. This pipeline has been evaluated using the EMBER open-source dataset, comprising roughly 1 million PE samples, which were statically analyzed. The obtained malware detection outcomes align with those from other academic studies within the current state of the art. Additionally, a comprehensive classification of malicious samples has also been performed. The models employed in this pipeline yield understandable results, aiding security analysts in comprehending the decisions made by the automated process [37].

M. Asam *et. al.* have introduced two fresh approaches for malware classification: the Deep Feature Space-based Malware Classification (DFS-MC) and the Deep Boosted Feature Space-based Malware Classification (DBFS-MC). In the DFS-MC framework, deep features are derived from tailored CNN architectures and are given as input to a SVM for malware classification. Conversely, in the DBFS-MC framework, enhanced discrimination capability is achieved by integrating deep feature spaces from two customized CNN architectures to create amplified feature spaces. Furthermore, the identification of unusual malware involves employing the deep boosted feature space with SVM. The efficacy of these frameworks is assessed using the MalImg dataset and the hold-out cross-validation method. The proposed DBFS-MC demonstrates improved performance in correctly classifying intricate malware types by leveraging feature boosting created through tailored CNNs. Notably, the DBFS-MC classification framework exhibits favorable results in F-score (0.96), accuracy (98.61%), recall (0.96) and precision (0.96) when tested rigorously using 40% previously unseen data [38].

Conventionally, antimalware solutions rely on signatures to detect known malware. However, this method encounters limitations in identifying obfuscated and packed malware effectively. Recognizing that understanding a program's structural aspects, such as mnemonics, instruction opcodes and API calls, often reveals the root of an issue, M. Ashik *et. al.* have explored the relevance of these features in distinguishing between unpacked malicious and benign executables in their work. Notably, significant features are gathered using Minimum Redundancy and Maximum Relevance (mRMR) techniques and Analysis of Variance (ANOVA). The study uses four datasets for experimentation employing ML and DL methods, including SVM, NB, J48, RF and XGBoost. Additionally, the performance evaluation involves an assortment of DNN, such as Deep Dense networks, CNN-LSTM and 1D-CNN to classify unknown samples, demonstrating promising outcomes particularly with system and API calls. The combination of system/API calls with static features marginally improves performance compared to models solely trained on dynamic features. Furthermore, to enhance accuracy, distinct deep learning methods have been implemented, showcasing a fine-tuned DNN that yields an F1-measure of 99.1% on dataset-2 and F1-measure of 98.48% on Dataset-3 [39].

G. Ahn *et. al.* have employed a machine learning algorithm to attain a detection accuracy exceeding 99% for identifying malicious files in their work. Additionally, a technique for visualizing data using the dynamic-analysis-based MITRE ATT&CK framework have been introduced by authors for malicious file detection. The PE malware dataset underwent classification utilizing Random Forest, Adaboost and Gradient Boosting models. These models demonstrated accuracies of 99.3%, 98.4% and 98.8%, respectively. The analysis of malicious file behavior was derived by visualizing the data through the application of the MITRE ATT&CK matrix [40].

S. Aurangzeb *et. al.* have demonstrated the value of utilizing a hardware execution profile to unveil the true execution landscape, aiding in the identification of obfuscated ransomware. Authors have assessed the efficacy of features extracted from hardware performance counters in categorizing malignant applications into ransomware and non-ransomware groups, employing various ML algorithms like RF, DT, GB and Extreme GB. The dataset used consists of 80 ransomware applications and 80 non-ransomware applications sourced from the VirusShare platform. The outcomes has highlighted the significant role of extracted hardware features in effectively identifying and detecting ransomware, achieving an F1-score of 0.97 with RF and Extreme GB [41].

J. Hemalatha *et. al.* have employed a visualization-based technique, representing malware binaries as 2D-images and a DL model for classification. The proposed malware classification system, based on DL, has implemented a reweighted class-balanced loss function within the last classification layer of the DenseNet model. This adaptation significantly enhances performance in classifying malware by addressing imbalanced data concerns. Extensive experiments have been conducted across four standard malware datasets. The results demonstrate that the proposed technique excels in detecting new malwares with heightened accuracy (98.46% for the BIG 2015 dataset, 98.23% for the Malimg dataset, 98.21% for the

MaleVis dataset, and 89.48% for the unseen Malicia dataset). Furthermore, it diminishes false-positive occurrences compared to traditional malware detection methods while preserving efficient computational time. Notably, the proposed malware detection method remains effective and reliable against obfuscation attacks [42].

To tackle the issue of real-time zero-day malware detection, Z. He *et. al.* introduced an ensemble learning-based technique. This approach aims to enhance the efficacy of conventional malware detectors, even when relying on a constrained set of micro-architectural features obtained in real-time from existing HPCs. Experimental results showcase that the proposed approach, employing AdaBoost ensemble learning on the RF classifier as the primary classifier, achieves impressive results. Specifically, it attains a 92% F-measure and a 95% True Positive Rate (TPR) while maintaining only a 2% false positive rate in detecting zero-day malware, leveraging solely the top 4 micro architectural features [43].

In Table2, all the techniques have been presented along with important features which have been extracted from original study.

Table2: ML based Malware Detection Techniques for Windows Platform

Ref.	Year	ML Models used for Classification	Dataset	Dataset Availability	Features Extracted from	Model with best performance	Approach Used	Analysis Type
[1]	2019	SVM, DT, RF, NB classifier	Virusshare and Vxhaven websites	Public	PE Headers	RF with Accuracy: 98.63%	Feature based Approach	Static
[14].	2022	RF, SVM, DT, AdaBoost, GNB, Gradient Boosting	Kaggle and Malware dataset from github	Public	PE Headers	RF with Accuracy: 99.44%	Feature based Approach	Static
[15]	2022	KNN, SVM, RBF SVM, RF, AdaBoost, DT, ExtraTrees, Linear Discriminant Analysis, Logistic, Neural Net, Passive Classifier, Ridge Classifier and SGD classifier.	ClaMP	Public	PE Headers	ExtraTrees with Accuracy: 98.8%	Feature based Approach	Static
[5]	2021	ResNet-50, AlexNet, CNN	Malimg, Microsoft BIG 2015, Malevis	Public	Windows PE Files	Accuracy: 97.78%	Feature based Approach	Static
[22]	2022	CNN, SVM, DT	dataset provided by the Canadian Institute for Cybersecurity	Private	PE Files	DT with Accuracy: 99%	Feature based Approach	Static
[24]	2022	CNN-LSTM	Kaggle Microsoft	Public	API Calls	Accuracy: 99%	Feature based Approach	Static
[25]	2023	Transfer Learning using Inception-V3	BIG15 dataset by Microsoft	Public	PE Files	Accuracy: 98.76%	Image based Approach	Static
[29]	2022	RF, KNN, XGBoost, LR	Self Collected	Private	Byte Code Files	RF and XGBoost are Better	Feature based Approach	Static
[30]	2022	RF with TLSH	VirusShare, VirusTotal, theZoo, IEEE, Malwr, Lenny Zelter and Contagio malware repositories	Public	PE Files, API Calls	Accuracy: 99.8%	Feature based Approach	Static
[34]	2022	CNN	Self Collected	Private	PE Files	Accuracy: 95%	Image based Approach	Static
[35]	2022	AdaBoost DT	Evasive-PDFMal2022	Public	PDF Files	Accuracy: 98.84%	Feature based Approach	Static
[36]	2021	Ensemble of Dense ANN and 1-D CNN with extra trees	Kaggle	Public	Windows PE Files	Accuracy: 100%	Feature based Approach	Static

[37]	2021	Pipeleine based Gradient Boosting Decison Tree (GBDT)	EMBER	Public	Windows PE Files	TPR: 86.3%	Feature based Approach	Static
[38]	2021	Customized CNN, SVM, ResNet-18 and DenseNet-201	MalImg malware dataset	Available on Request	PE Files	Accuracy: 98.61%	Feature based Approach	Static
[42]	2021	DenseNet	MalImg, BIG 2015, MaleVis, Malicia	Public	PE Binary Files	Accuracy: MalImg: 98.23%, BIG 2015: 98.46%, MaleVis: 98.21%, unseen Malicia: 89.48%	Image based Approach	Static
[27]	2022	Customized CNN	VirusShare	Public	Hardware Performance Counters	Recall: 98.6%	Image based Approach	Dynamic
[28]	2022	Cost-Sensitive Pareto Ensemble classifier, Deep Contractive Autoencoder (DCAE)	Obtained from [44]	Available on Request	Windows API Calls	Recall: 99%	Feature based Approach	Dynamic
[33]	2022	NB, KNN, SVM, RF, J48	Resilient Information System Security (RISS) Research Group	Available on Request	API Calls	SVM with Accuracy: 97.05	Feature based Approach	Dynamic
[41]	2021	RF, DT, GB, Extreme GB	VirusShare	Public	Hardware performance Counter	RF and ExGB with F1-Score: 0.97	Feature based Approach	Dynamic
[43]	2021	Adaboost Ensemble Learning, Random Forest	VirusTotal, VirusShare	Public	Hardware Performance Counter	F1-Score: 92%	Feature based Approach	Dynamic
[32]	2022	RF, Neural Network, KNN, SVM, DT, NB	IEEE DataPort	Public	PE Files, API Calls	RF with Accuracy: 98.92%	Feature based Approach	Both (Static + Dynamic)
[26]	2022	XGBoost, ET	VirusShare, PortableFreeware, PortableApps	Public	PE Files	Accuracy: 91.9% (static), 96.4% (dynamic)	Feature based Approach	Static+Dynamic
[23]	2023	SVM, CNN	Malimg	Public	PE Files	Accuracy: 99.06%	Feature based Approach	Hybrid

5. Evaluation and Discussion

5.1 Overview of the Key Findings

In Table 2, the different ML based techniques for malware detection on windows platform have been presented based on selected factors including ML model used, dataset used, performance of model, Type of files from where features were extracted and approach type. In this section, we will discuss and give an evaluation of these approaches.

Comparison of ML Models:

1. Random Forest (RF) seems to be the best choice across studies, as it is showing highest accuracy in multiple research works (98.63%, 99.44%, 98.8%, 99.18%, 99.37%, 99.8%, 99.06%, 99.8992%, 99%, 99.8%, 99.06%, etc.).
2. Other than RF model, ExtraTrees, XGBoost, SVM, Decision Trees (DT), CNN, AdaBoost have also shown notable accuracy or F1-scores in the range of 97%-99.8% in different datasets.

Dataset Variations:

1. The datasets used by researchers vary from publicly available datasets to self-collected or private datasets.
2. Most of the studies have used publically available dataset for experimentation and performance evaluation of their work.
3. The dataset choice seems to influence the performance of model, with some models giving high accuracy on specific datasets but giving lower accuracy on unseen datasets.

Feature-based vs. Image-based Approaches:

1. Both feature-based and image-based approaches have shown effectiveness in malware detection.
2. Most of the research has been done on feature-based methods and features have been extracted from PE headers, API calls, Windows PE files and hardware performance counters.

3. Only few researchers have worked on Image-based methods using images of API calls or PE files.

Static vs. Dynamic Analysis:

1. Static Analysis examines the code or structure of a file without executing it. It doesn't require execution of file, which makes it faster and less resource-intensive. It can detect known patterns of malware. But it is challenging to detect obfuscated malware using static analysis.
2. Dynamic Analysis involves executing the file in a controlled environment to observe its behavior. It provides valuable insight into the actual behavior of malware, which helps indetection of new or unseen threats. It is slower and more resource-intensive compared to static analysis.
3. Both methods have their strengths and limitations. The overall effectiveness of malware detection can be enhanced by forming a hybrid technique with integration of static and dynamic analysis techniques.

Private vs. Public Datasets:

1. Studies using private datasets have reported high accuracy, possibly due to customized data collections.
2. Public datasets are commonly used but may not capture the diverse real-world malware, impacting overall adoptability of model.

5.2 Summary of the Key Contributions

From our extensive review of literature on machine learning-based malware detection for Windows systems, this review brings forth following key contributions:

Systematic Literature Review:

This paper systematically examines the relevant literature on machine learning methods to detect malware on Windows systems. The papers were selected based on specific criteria and a comprehensive analysis was conducted.

An Overview of Machine Learning based Windows Malware Detection Techniques:

The review covers a range of ML algorithms and datasets employed in detecting Windows malware. This information can be valuable for researchers and professionals aiming to understand the current state of the art in this area.

Future Directions:

Future directions for ML based malware detection on Windows platform has been highlighted in this review. The review proposes ideas to enhance current methods and develop more efficient approaches for this purpose.

By presenting a thorough evaluation of the current state of the research, assessing strengths and weaknesses of existing methods and outlining problems for future research, this review significantly contributes to the domain of ML based windows malware detection. The findings of this paper can guide researchers to develop more effective and efficient detection systems and to contribute towards advancement of future research in this domain.

6. Future Research Directions

- The growing adoption of ML based techniques for malware detection has led to a lot of research in this area. The research community has proposed different techniques to detect malware at an early stage with an effective accuracy. However, there are still some problems which need to be addressed in order to improve detection accuracy. The following are some future research directions for windows malware detection based on the findings of this review;
- The research can be done to study the impact of using a hybrid ML model formed by integrating different models. The use of hybrid model can improve the classification performance.
- The overall effectiveness of malware detection can be enhanced by forming a hybrid approach with integration of static and dynamic analysis.
- There is a need to develop a generalized model which could be able to detect different types of malware effectively and can also adapt to detect new, unseen malwares.
- The work can be done to develop a real time detection system to detect malware as they emerge in real time.
- There is a need to work on expanding the existing datasets and making them more diverse by adding different malware characteristics and behaviors.
- Most of the research is based on windows API calls and PE files which are not inherently security-focused. Future research in windows malware detection should delve deeper into security-oriented attributes like permission requests and system logs.

In summary, there is considerable space for more research in the domain of ML-based windows malware detection. The future research directions from this review aim to advance the field forward, improve the efficiency of windows malware detection techniques and offer a valuable starting point for future studies.

7. Conclusion

The widespread use of Windows has drawn the attention of malicious actors looking to take advantage of its popularity. Windows malware poses a significant risk to the security of windows platform and its users. Therefore, Detecting windows malware has become a crucial research domain. Several Machine learning based solutions have been presented and implemented by the research community to solve this critical problem. In this paper, we have performed a comprehensive literature review to explore the different ML based techniques to detect the windows malware. Our aim was to provide an in-depth understanding of the current state of the research in this domain, highlight key findings from recent related studies and its limitations and suggesting potential future research problems which still need to be

addressed in order to detect malware at an early stage efficiently.

In conclusion, this paper presents a comprehensive review of the current landscape of windows malware detection using ML. The significant findings and contributions outlined in this survey offer valuable insights to researchers. Furthermore, by outlining limitations and suggesting future research directions, this study provides a roadmap for future studies in this field. We consider that this paper will serve as a valuable resource for the researchers working in this area.

Support Vector Machine:	SVM
Decision Tree:	DT
Logistic Regression:	LR
Random Forest:	RF
Naive Bays:	NB
Gaussian Naive Bayes:	GNB
Radial Basis Function:	RBF
Portable Executeable:	PE
Stochastic Gradient Descent:	SGD
Extreme Gradient Boosting:	XGBoost
Extreme Random Trees:	ET
Stochastic Gradient Boosting:	SGB

References

[1] S. Naz and D. K. Singh, "Review of machine learning methods for windows malware detection," in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6, 2019.

[2] J. Singh and J. Singh, "A survey on machine learning-based malware detection in executable files," Journal of Systems Architecture, vol. 112, pp. 101861, 2021.

[3] M. N. U. R. Chowdhury, A. Haque, H. Soliman, M. S. Hossen, T. Fatima, and Ahmed, I., "Android malware Detection using Machine learning: A Review," arXiv preprint arXiv:2307.02412, 2023.

[4] Ö. Aslan and R. Samet, "A comprehensive review on malware detection approaches," IEEE Access, vol. 8, pp. 6249-6271, 2020.

[5] Ö. Aslan and A. A. Yilmaz, "A new malware classification framework based on deep learning algorithms," IEEE Access, vol. 9, pp. 87936-87951, 2021.

[6] Q. Wu, X. Zhu, and B. Liu, "A survey of android malware static detection technology based on machine learning," Mobile Information Systems, vol. 2021, pp. 1-18, 2021.

[7] E. J. Alqahtani, R. Zagrouba, and A. Almuhaideb, "A survey on android malware detection techniques using machine learning algorithms," in 2019 Sixth International Conference on Software Defined Systems (SDS), pp. 110-117, 2019.

[8] V. Kouliaridis and G. Kambourakis, "A comprehensive survey on machine learning techniques for android malware detection," Information, vol. 12, no. 5, pp. 185, 2021.

[9] J. Senanayake, H. Kalutarage, and M. O. Al-Kadri, "Android mobile malware detection using machine learning: A systematic review," Electronics, vol. 10, no. 13, pp. 1606, 2021.

[10] M. Al-Janabi and A. M. Altamimi, "A comparative analysis of machine learning techniques for classification and detection of malware," in 2020 21st International Arab Conference on Information Technology (ACIT), pp. 1-9, 2020.

[11] J. Qiu, J. Zhang, W. Luo, L. Pan, S. Nepal, and Y. Xiang, "A survey of android malware detection with deep neural models," ACM Computing Surveys (CSUR), vol. 53, no. 6, pp. 1-36, 2020.

[12] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A review of android malware detection approaches based on machine learning," IEEE Access, vol. 8, pp. 124579-124607, 2020.

[13] Z. Wang, Q. Liu, and Y. Chi, "Review of android malware detection based on deep learning," IEEE Access, vol. 8, pp. 181102-181126, 2020.

[14] A. Hussain, M. Asif, M. B. Ahmad, T. Mahmood, and M. A. Raza, "Malware detection using machine learning algorithms for windows platform," in Proceedings of International Conference on Information Technology and Applications: ICITA, Singapore: Springer Nature Singapore, pp. 619-632, 2022.

[15] R. Damaševičius, A. Venčkauskas, J. Toldinas, and Š. Grigaliūnas, "Ensemble-based classification using neural networks and machine learning models for windows pe malware detection," Electronics, vol. 10, no. 4, pp. 485, 2021.

[16] M. Almousa, S. Basavaraju, and M. Anwar, "Api-based ransomware detection using machine learning-based threat detection models," in 2021 18th International Conference on Privacy, Security and Trust (PST), pp. 1-7, 2021.

[17] A. Irshad, R. Maurya, M. K. Dutta, R. Burget, and V. Uher, "Feature optimization for runtime analysis of malware in windows operating system using machine learning approach," in 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), pp. 255-260, 2019.

[18] F. O. Catak, A. F. Yazı, O. Elezaj, and J. Ahmed, "Deep learning based Sequential model for malware analysis using Windows exe API Calls," PeerJ Computer Science, vol. 6, pp. e285, 2020.

[19] X. Huang, L. Ma, W. Yang, and Y. Zhong, "A method for Windows malware detection based on deep learning," Journal of Signal Processing Systems, vol. 93, pp. 265-273, 2021.

[20] D. Rabadi and S. G. Teo, "Advanced Windows methods on malware detection and classification," in Annual Computer Security Applications Conference, pp. 54-68, 2020.

[21] K. Sethi, R. Kumar, L. Sethi, P. Bera, and P. K. Patra, "A novel machine learning based malware detection and classification framework," in 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1-4, 2019.

[22] M. S. Akhtar and T. Feng, "Malware Analysis and Detection Using Machine Learning Algorithms," Symmetry, vol. 14, no. 11, pp. 2304, 2022.

[23] K. Shaukat, S. Luo, and V. Varadharajan, "A novel deep learning-based approach for malware detection," Engineering Applications of Artificial Intelligence, vol. 122, pp. 106030, 2023.

[24] M. S. Akhtar and T. Feng, "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time," Symmetry, vol. 14, no. 11, pp. 2308, 2022.

[25] M. Ahmed, N. Afreen, M. Ahmed, M. Sameer, and J. Ahamed, "An inception V3 approach for malware classification using machine learning and transfer learning," International Journal of Intelligent Networks, vol. 4, pp. 11-18, 2023.

[26] J. Palša, N. Ádám, J. Hurtuk, E. Chovancová, B. Madoš, M. Chovanec, and S. Kocan, "Mlmd—a malware-detecting antivirus tool based on the xgboost machine learning algorithm," Applied Sciences, vol. 12, no. 13, pp. 6672, 2022.

[27] G. O. Ganfure, C. F. Wu, Y. H. Chang, and W. K. Shih, "Deepware: Imaging performance counters with deep learning to detect ransomware," IEEE Transactions on Computers, vol. 72, no. 3, pp. 600-613, 2022.

[28] U. Zahoora, A. Khan, M. Rajarajan, S. H. Khan, M. Asam, and T. Jamal, "Ransomware detection using deep learning based unsupervised feature extraction and a cost-sensitive Pareto Ensemble classifier," Scientific Reports, vol. 12, no. 1, pp. 15647, 2022.

[29] P. Tumuluru, L. R. Burra, M. V. V. Reddy, S. Sudarsa, Y. Sreeraman, and A. L. A. Reddy, "APMWMM: Approach to Probe Malware on Windows Machine using Machine Learning," in 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp. 614-619, 2022.

[30] M. Kumar, "Scalable malware detection system using big data and distributed machine learning approach," Soft Computing, vol. 26, no. 8, pp. 3987-4003, 2022.

[31] S. S. Alshamrani, "Design and Analysis of Machine Learning Based Technique for Malware Identification and Classification of Portable Document Format Files," Security and Communication Networks, vol. 2022, 2022.

[32] F. Alhaidari, N. A. Shaib, M. Alsafi, H. Alharbi, M. Alawami, R. Aljindan, and R. Zagrouba, "ZeVigilante: detecting zero-day malware using machine learning and sandboxing analysis techniques," Computational Intelligence and Neuroscience, vol. 2022, 2022.

[33] W. Z. Zakaria, M. F. Abdollah, O. Mohd, S. W. M. S. M. Yassin, and A. Ariffin, "RENTAKA: A Novel Machine Learning Framework for Crypto-Ransomware Pre-encryption Detection," International Journal of Advanced Computer Science and Applications, vol. 13, no. 5, pp. 378-385, 2022

[34] E. V. P. Kalyan, A. P. Adarsh, S. S. L. Reddy, and P. Renjith, "Detection of malware using CNN," in 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1-6, 2022.

[35] Q. Abu Al-Haija, A. Odeh, and H. Qattous, "PDF malware detection based on optimizable decision trees," Electronics, vol. 11, no. 19, pp. 3142, 2022.

[36] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE malware detection using ensemble learning," Informatics, vol. 8, no. 1, pp. 10, 2021.

[37] N. Loi, C. Borile, and D. Ucci, "Towards an automated pipeline for detecting and classifying malware through machine learning," arXiv preprint arXiv:2106.05625, 2021.

[38] M. Asam, S. J. Hussain, M. Mohatram, S. H. Khan, T. Jamal, A. Zafar, and U. Zahoora, "Detection of exceptional malware variants using deep boosted feature spaces and machine learning," Applied Sciences, vol. 11, no. 21, pp. 10464, 2021.

[39] M. Ashik, A. Jyothish, S. Anandaram, P. Vinod, F. Mercaldo, F. Martinelli, and A. Santone, "Detection of malicious software by analyzing distinct artifacts using machine learning and deep learning algorithms," Electronics, vol. 10, no. 14, pp. 1694, 2021.

[40] G. Ahn, K. Kim, W. Park, and D. Shin, "Malicious file detection method using machine learning and interworking with MITRE ATT&CK framework," Applied Sciences, vol. 12, no. 21, pp. 10761, 2022.

[41] S. Aurangzeb, R. N. B. Rais, M. Aleem, M. A. Islam, and M. A. Iqbal, "On the classification of Microsoft-Windows ransomware using hardware profile," PeerJ Computer Science, vol. 7, pp. e361, 2021.

[42] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for malware detection," Entropy, vol. 23, no. 3, pp. 344, 2021.

[43] Z. He, T. Miari, H. M. Makrani, M. Aliasgari, H. Homayoun, and H. Sayadi, "When machine learning meets hardware cybersecurity: Delving into accurate zero-day malware detection," in 2021 22nd International Symposium on Quality Electronic Design (ISQED), pp. 85-90, 2021.

[44] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," arXiv preprint arXiv:1609.03020, 2016.

Information for Authors

Submission: Manuscripts [in Word (.doc, .docx, .rtf)] should be submitted by one of the authors of the manuscript through the online submission system at www.thencleuspak.org.pk after registration of corresponding author. If for some technical reason on-line submission is not possible, then write an email describing the problem along with your phone no. at editorinchief@thenucleuspak.org.pk

Terms of Submission: Each submission to The Nucleus implies that the manuscript presents the results of original scientific research and has not been published nor has been submitted for publication elsewhere. The article is written in clear and Standard English. The reported research meets all applicable ethical standards and research integrity. The submitted manuscripts is screened for plagiarism during the editorial process.

Units of Measurement: should be presented simply and concisely using System International (SI) units.

Article Structure

Subdivision - numbered sections: The article should be divided into clearly defined and numbered sections. Subsections should be numbered 1.1 (then 1.1.1, 1.1.2, ...), 1.2, etc. (the abstract is not included in section numbering). Any subsection may be given a brief heading. Each heading should appear on its own separate line.

Title: should be concise and informative. Avoid abbreviations and formulae where possible.

Author names and affiliations: Provide complete name of all the authors, their affiliation, complete postal addresses, contact numbers and e-mail addresses. Present the authors' affiliation addresses below the names. Indicate all affiliations with a lower-case superscript letter immediately after the author's name. Clearly indicate the corresponding author by superscript*. Further when manuscript is under review process, as per policy of the journal, author cannot be addeded, deleted and sequence of author can't be altered.

Keywords: Provide a maximum of 6 keywords, These keywords will be used for indexing purposes.

Introduction Section: States the objectives of the work and provides an adequate background, avoiding a detailed literature survey or a summary of the results.

Experimental Section: should contain sufficient detail to allow the work to be reproduced. Methods already published should be indicated by a reference, only relevant modifications should be described.

***Theory/calculation S**ection:* Should extend, not repeat, the background to the article already dealt with in the Introduction and lay the foundation for further work. In contrast, a Calculation section represents a practical development from a theoretical basis.

Results and Discussion: should provide the significance of the results of the work. A combined Results and Discussion section is often appropriate. Avoid extensive citations and discussion of published literature.

Conclusions: It should be presented in a short Conclusions section.

Acknowledgments: (if any) should be included at the very end of the paper before the references and may include supporting grants, presentations, and so forth.

References: We follow IEEE style for citations of references. Must be numbered consecutively and citations of references in text should be identified using numbers in square brackets (e.g., as discussed by Smith [9]; as discussed elsewhere [9, 10]). Reference to a publication:

[Ref number] Author's initials. Author's Surname, "Title of article," Title of journal abbreviated in Italics, vol. number, issue number, page numbers, Abbreviated Month Year.

[4] K.A. Nelson, R.J. Davis, D.R. Lutz, and W. Smith, "Optical generation of tunable ultrasonic waves," Journal of Applied Physics, vol. 53, no. 2, pp. 1144-1149, Feb., 2002.

For more details please see the link IEEE style for citations of different materials

Figures and Tables: Include all figures and tables in the word file of the manuscript. Figures and tables should not be submitted in separate files. If the article is accepted, authors may be asked to provide the source files of the figures. All figures should be cited in the paper in a consecutive order. In all figures, remove all unnecessary boxes, lines, marks. The resolution of all the figures must be at least 300 dpi. Tables should be cited consecutively in the text. Every table must have a descriptive title and if numerical measurements are given, the units should be included in the column heading. Vertical rules should not be used.

The Nucleus

An Open Access International Scientific Journal

ISSN : 0029-5698 (Print)
EISSN : 2306-6539 (Online)

Recognized by HEC in 'Y' Category

Call for Papers

Why Publish in The Nucleus?

- One of the Oldest Scientific Journals in Pakistan
- Regularly Published since 1964
- Published Both Electronically & in Paper Format
- Multidisciplinary
  - Natural Sciences
  - Applied Sciences
  - Engineering & Technology
  - Management Sciences
- Open Access
- Peer Reviewed*
- No Publication Charges
- High Visibility
- Electronic Submission
- Rapid On-line Publication (within three months)

Abstracted and Indexed in:

- Chemical Abstracts
- Biological Abstracts
- INIS Atom Index
- Bibliography of Agriculture (USA)
- The Institute of Electrical Engineers Publications
- Virology Abstracts (England)
- Pakistan Science Abstracts

For Further Information

Editorial Office The Nucleus
PINSTECH, 45650 Nilore
Islamabad. Pakistan
For Online
http://www.thenucleuspak.org.pk
E-mail
editorialoffice@thenucleuspak.org.pk

Why Perish when you can Publish in The Nucleus?

*Potential Reviewers are Invited to Submit their CV's Through E-mail