

Analysis of Enhanced Adaptive Gradient Boosting Regression in Business Intelligence

Mr. Kopelo Letou¹, Dr. S.Thiyagarajan²

^[1] Computer Science and Engineering, St. Joseph University, Chümoukedima, Nagaland, India

^[2] Computer Science and Engineering, St. Joseph University, Chümoukedima, Nagaland, India.

^[1] kopsystem@gmail.com, ^[2] s.thiyagarajin@gmail.com

Abstract— Enhanced Adaptive Gradient Boosting (EAGB) is a real-time, streaming version of Gradient Boosting that updates weak learners' one sample at a time and adapts automatically to concept drift through continuous error monitoring and dynamic model adjustments. It extends traditional gradient boosting by using incremental Hoeffding Tree regressors as base learners and updating the model continuously as new data arrive, rather than relying on static batch training. EAGB is particularly well-suited for Business Intelligence and forecasting tasks, including revenue prediction, demand forecasting, and financial analytics in streaming environments. Enhanced Adaptive Gradient Boosting Regression is important because it delivers consistently high predictive accuracy while effectively modeling complex and non-linear data patterns commonly found in real-world applications such as e-commerce, finance, healthcare, retail and others. The purpose of Business Intelligence is to attain desire's optimum goal in business, rapid business decision making while minimizing time, resources consumption, and carbon footprint emission. The Enhanced Adaptive Gradient Boosting Regression in Business Intelligence helps organizations turn data into highly accurate predictions and insights, enabling better decisions and quicker responses. Enhanced Adaptive Gradient Boosting Regression in Business Intelligence serves vital role in addressing the real-world problems of businesses for example in retail to determine the sales of goods, revenue and services to consumers. Analyzing the Enhanced Adaptive Gradient Boosting Regression in Business Intelligence (BI) is crucial as it allows the organization to manage the trade-off between model performance, accuracy, cost, efficiency, resources consumption like CPU, RAM and emission of carbon dioxide from the computer's system. The mechanism of Enhanced Adaptive Gradient Boosting Regression in Business Intelligence integrates sequential gradient boosting with online Hoeffding trees, residual clipping, EMA-based baseline updates, and ADWIN drift detection to enable continuous learning and incremental error correction. This design allows the model to adapt quickly to changing revenue patterns while maintaining stability during fluctuations, resilience to abrupt concept drifts, and efficiency in real-time processing of data. The novel aspect of this paper was that we developed a Business Intelligence model using Enhanced Adaptive Gradient Boosting Regression to solve the real-world business problems with accurate, quick response as well as better decision making in business. And also, we have analysed this model which compute performance metrics such as MSE, MAE, R2, and SMAPE, estimated system's resources CPU, RAM usage, training time, energy consumption and estimated CO₂ emissions.

Index Terms— Business Intelligence (BI), Enhanced Adaptive Gradient Boosting (EAGB) Regression, Gradient Boosting (GB), Green Business Intelligence (GBI).

1. INTRODUCTION

Businesses as the process of making decisions to generate maximum return on investment of the resources allocated to various products. According to (Kopelo Letou and S.Thiyagarajan., 2023), businesses difficulties can be address by shifting from

outdated or traditional Business Intelligence (BI) approaches to Adaptive BI models that use machine learning (ML) algorithms and Artificial Intelligence (AI) approaches. Machine learning technology in BI is essential to resolve real-world business problems, resolve business decisions quickly, improve customer and firm service, automate operational business issues, adapt quickly to market trends without delay, estimate more accurately, predict business operations more precisely, and maximize business profits [1]. Many researchers have defined BI from different perspectives. BI is an adaptive business intelligence model which includes processes, products, components, ML technology that extract business data from huge databases as well as transform it into intelligence for decision-making, problem-solving, achieving optimum business goals, controlling its actions. BI is a broad concept, discipline, and set of technologies used for transforming raw data into meaningful insights which assist decision-making. BI is method of transforming business data into intelligence for decision making in obtaining optimal desired goals. BI is the overall system and process for turning data into insights. BI is an adaptive business intelligence model which includes processes, products, components, ML technology that extract business data from huge databases and transform it into intelligence for decision-making, problem-solving, achieving optimum business goals, controlling its actions. BI continues to be defined by the processes of acquiring, processing, and utilizing data to make better business decisions. A BI model is the framework, design, or structure that defines how BI is implemented within an organization. A BI model is the blueprint for how BI works internally. In context of ML, BI model used Enhanced Adaptive Gradient Boosting (EAGB) Regression algorithm is a real-time, streaming version of Gradient Boosting (GB) that updates weak learners' one sample at a time and adapts automatically to concept drift through continuous error monitoring and the model updates incrementally to predict continuous values. Adaptive Business Intelligence (GBI) is described as adaptive business intelligence model involving products, components, processes, ML techniques to transform raw data into actionable intelligence for optimal decision-making while minimizing time, resource consumption, and carbon footprint emission. BI is important because it helps organizations turn raw data into actionable insights which assist better decision-making. It improves how a business plans, operates, competes, and grows. According to (Kopelo Letou and S.Thiyagarajan., 2023), main purpose of BI Machine is to attain desire's optimum goal in business, rapid business decision making while minimizing time, resources consumption, and carbon footprint emission. Utilities of BI are in retail, healthcare, finance, healthcare, pharmaceutical industry, et cetera [1]. There is no single, universally standardized definition of Enhanced Adaptive Gradient Boosting. EAGB is a real-time, streaming version of Gradient Boosting that updates weak learners' one sample at a time and adapts automatically to concept drift through continuous error monitoring and dynamic model adjustments. Enhanced Adaptive Gradient Boosting Regression is advanced online ensemble machine learning approach intended for regression tasks in dynamic and streaming data environments. It extends traditional gradient boosting by using incremental Hoeffding Tree regressors as base learners and updating the model continuously as new data arrive, rather than relying on static batch training. EAGB learns by sequentially fitting trees to residuals (gradients) of a loss function, allowing every tree to correct errors of preceding ones in real time. What makes EAGB 'enhanced' is its ability to adapt to changing data distributions through components such as ADWIN drift detection, residual clipping for robustness, and exponential moving averages (EMA) for tracking baseline trends. These mechanisms enable the model to remain stable under noise, resilient to concept drift, and efficient for large-scale, real-world applications. As a result, EAGB is particularly well-suited for BI

and forecasting tasks, including revenue prediction, demand forecasting, and financial analytics in streaming environments. GB is ensemble ML approach which generates strong learner through integrating weak learners to improve dataset prediction. Another definition of Gradient Boosting is an algorithm that develops model gradually or consecutively, with each new model attempting to use gradient descent to eliminate the errors (residuals) of the preceding models. To minimize a loss function, gradient boosting integrates additional trees pointing in direction of steepest descent using gradient descent. It is named 'Gradient' as algorithm determines how to update the model based on gradient of loss function. Gradient Boosting is called "gradient" because it iteratively proceeds in direction of steepest descent (negative gradient) in model's parameter space to reduce loss function (namely Mean Squared Error for regression or log-loss for classification). GB is called 'Boosting' because it boosts performance through integrating many weak learners into one strong model. The term "Gradient Boosting" refers to method of minimizing the model's loss function (error) by gradient descent. Instead of directly fitting the weak learner to the output variable y , it fits it to negative gradient of loss function with respect to current ensemble's prediction. GB builds the final model by sequentially adding weak learners, where every new model tries to reduce errors of previous combined ensemble —this method is known as boosting. (Mahesh Batta.2018), According to Arthur Samuel, ML is area of research which enables computers to learn without explicit programming [16]. According to Pega Documentation (2025), Adaptive Gradient Boosting is an online learning algorithm that achieves higher predictive power [17]. EAGB is a real-time, streaming version of GB that updates weak learners one sample at a time and adapts automatically to concept drift through continuous error monitoring and dynamic model adjustments. The 3 types of Gradient Boosting (GB) are GB Regression, GB Classification and GB Ranking. GB Regression is procedure which predicts continuous numerical values. For example, predicting stock prices, sales forecasting and house prices. GB Classification is approach which predicting categorical or class labels. For example, spam detection, fraud detection and others. GB Ranking is a technique that predicts the order or relevance or importance of items, for example search result ranking or recommendation systems. EAGB Regression is important because it delivers consistently high predictive accuracy while effectively modeling complex and non-linear data patterns commonly found in real-world applications such as e-commerce, finance, healthcare, retail, etc. It scales efficiently to large datasets, handles missing and noisy data robustly, and supports continuous learning in dynamic environments. The EAGB regression algorithm provides a robust and scalable framework for BI by integrating online Hoeffding trees, sequential gradient boosting, residual clipping, EMA-based baseline updates, and ADWIN drift detection. This combination enables the model to learn incrementally, adapt quickly to concept drift, maintain stability during fluctuations, and deliver reliable real-time predictions, making it well-suited for streaming and BI systems. The EAGB Regression in BI is essential in Green Business Intelligence (GBI) because it helps organizations turn data into highly accurate predictions and insights, enabling better decisions and quicker responses. EAGB Regression in BI plays as vital component in addressing the real-world constraints of businesses for example in retail to determine the sales of goods and services to consumers. It is a technique to enhance their decision-making and improve business outcomes in the areas of finance, healthcare, marketing, retail, etc. by providing approximately accurate predictions and deep insights. In other words, EAGB regression in BI is Green Business Intelligence model employs the EAGB technique which use data to analyze trends, predict and evaluate overall model performance, including computational resource utilization, energy consumption, carbon dioxide emissions, and

model training time. Green Business Intelligence model evaluates the trade-off between operational efficiency such as performance, resource utilization, and training time and environmental sustainability including energy consumption and carbon dioxide emissions. Green BI is not only about reports and dashboards it also increasingly relies on advanced analytics and machine learning. Analyzing the EAGB Regression in BI is crucial as it allows the organization to manage the trade-off between model performance, accuracy, efficiency, resources consumption like CPU, RAM and emission of carbon dioxide from the computer's system and environmental sustainability. BI systems must balance across performance of time, space, accuracy and energy consumption. Increase in global warming due to consumption of power and then produce carbon dioxide from industries and even from computer systems. It could pave way for integration of Green ICT, Green IT, Sustainable IT, ICT sustainability/green computing, a new revolution in environmentally sustainable computing, while reducing carbon footprint from computational resources utilized for computations and using ML and algorithmic efficiency to address real-world electronic business problems. The mechanism of Enhanced Adaptive Gradient Boosting Regression integrates sequential gradient boosting with online Hoeffding trees, residual clipping, EMA-based baseline updates, and ADWIN drift detection to enable continuous learning and incremental error correction. This design allows the model to adapt quickly to changing revenue patterns while maintaining stability during fluctuations, resilience to abrupt concept drifts, and efficiency in real-time processing. The novel aspect of this paper was that we developed a Business Intelligence model using Enhanced Adaptive Gradient Boosting Regression to solve the real-world business problems with accurate, quick response and better decision-making in business. And also, we have analysed this model that computes performance metrics such as MSE, MAE, R2, and SMAPE, estimated CPU, RAM usage, training time, system logs resource usage—including CPU load, RAM consumption, energy usage, and estimated CO₂ emissions.

II. RELATED WORK

Definition of business intelligence changes with time and space. There are different definitions of business intelligence (BI). According to (Kopelo Letou and S.Thiyagarajan., 2023; 2025), BI is a method of converting several types of business data into useful information and, eventually, intelligence which may support decision-making at every level and enhance business performance to achieve optimum goals. Term “business intelligence” in context of machine learning could be described as an adaptive BI model that includes procedures, products, components, ML technology that take business data from large databases and turn it into intelligence for decision-making, problem-solving, achieving optimal business goals, and controlling actions. The term “business” comprises more than just profit-making businesses; it also refers to application-based activities in any kind of organization, business, industry, or firm[1];[25]. According to (Davis Gary Alan and Woratschek Charles R., 2015), BI is defined as collecting, refining and acting upon information retrieved. Richard Millar Devens first used word “business intelligence” (BI) in 1865. The term “business intelligence” (BI) has been coined by R.M. Devens to discuss how banker made profit by obtaining and acting upon environmental information before their competitors [2]. Traditional BI systems are often static and non-adapt to new data and changing market dynamics. It's limiting their responsiveness in fast-changing business environments. Example - Microsoft Office (MS) tools, especially Excel, Statistical Analysis System Business Intelligence (SAS BI). Information is the backbone of business decision-making. (Gilad Benjamin and Gilad Tamar. 1986), defined BI is method of inputting raw data about the external business environment and

its end result is intelligence to make strategic decision-making. Business Intelligence activities include collection and evaluation of data reliability and validity, analysis of data, interpretation of data into intelligence and dissemination of intelligence [3]. (Friedman Jerome Hastie et al., 2000; Friedman Jerome H., 2001; Friedman Jerome H., 2002), describe gradient boosting. Regression uses least squares to fit simple parameterized function (base learner) to the current “pseudo”-residuals at every iteration to produce additive regression models. At each iteration, a subsample of the training data is chosen at random (without replacement) from the full training data set. The gradient of the loss functional being minimized in relation to the model values at each training data point assessed at the present step is known as the pseudo-residuals. Instead of using the full sample, this randomly selected subsample is used to fit the base learner and determine the model update for the current iteration [4,5,6]. Gradient Boost Algorithm is the backbone and foundation for Adaptive Gradient Boost and Enhanced Adaptive Gradient Boosting. Gradient Boosting sequentially fits trees to negative gradient (residuals) of loss function to correct errors of previous model. Each iteration tries to reduce the loss further by focusing on errors. The final prediction is an additive model combining all the trees. Learning rate ν controls how much each tree contributes, which prevents overfitting. Core principle of gradient boosting is correct-the-previous-error cycle. (Biau G., Cadre B., and Rouvière L., 2019), ML technique called gradient boosting regression integrates outcomes of numerous simple predictors to create an effective committee with enhanced performance than its individual members. The technique is known as gradient tree boosting and it is usually applied to decision trees of fixed size as base learner. Term “Gradient Boosting Regression” was first utilized by Friedman Jerome H. (2001, 2002)[7]. (Hosen Saikat Md and Amin Ruhul., 2021), ML approach for regression and cataloguing issues, gradient boosting regression forms prediction model in form of ensemble of weak prediction models. To put it another way, Gradient Boosting Regression develops a stronger model for prediction by combining groups of very weak prediction models. This algorithm is an effective method for developing prediction models [8]. (Mayr Andreas, Binder Harald, Gefeller Olaf and Schmid Matthias., (2014), Gradient Boosting Regression is ML technique that combines multiple examples into a more accurate prediction for enhancing accuracy of weak classification tool[9]. (Bahad Pritika and Saxena Preeti., 2019), ensemble ML approach known as gradient boosting regression produces high-accuracy model through merging multiple low-accuracy models. It can be used to improve prediction in a number of fields, including retail, credit, insurance, consumer behavior, and medical diagnosis [10]. (Chen Sihang, Zhang Na, Shui Biyuan, Zhang Tiantian, Li Chunman and Gong Qingjun., 2025), Gradient Boosting Regression is ensemble ML algorithm that minimizes differentiable loss function L by iteratively adding weak learners in direction of negative gradient of the loss, effectively performing gradient descent in function space [11]. (Nallakaruppan M.K., et al., 2025), Gradient Boosting Regression is effective ensemble learning method which systematically combines weak learners to minimize the loss function, creating a strong predictive model[12]. (Rafiq Islam., 2024), defines Gradient Boosting Regression as ensemble learning approach which generates strong learner through combining several weak learners, usually decision trees. The fundamental concept of boosting involves training models one after the other, with each new model attempting to fix errors of ones that came before it. This is achieved by gradient boosting regression, which uses gradient descent to minimize a loss function[13]. (Natekin Alexey and Knoll Alois KnollAlois., 2013), learning process in gradient boosting machines, or GBMs for short, fits new models one after another for generating more accurate estimate of response variable.

GBM, learning process is sequential and additive in nature. In this, first model gives a rough estimate. The second model focuses on what the first model got wrong. The third model corrects what the second model got wrong and so on. Finally, the sum of all corrections leads to a powerful community prediction model [14]. Drawback of generic GB was that data processing was batch training on a fixed dataset and making it unsuitable for streaming or real-time updating model incrementally. There is no built-in mechanism for concept drift detection or adaptation, so the model cannot react to changes in data distributions over time. Each tree is typically trained once per boosting iteration in a deterministic manner, limiting learning flexibility. The generic GB model starts with a fixed constant baseline and does not support adaptive baseline updates during training. Additionally, residuals were not explicitly clipped, which can lead to instability when large errors occur. Finally, trees were trained sequentially without any capability for dynamic resetting, pruning, or replacement, reducing robustness in non-stationary environments. In this regards, we have proposed Enhanced Adaptive Gradient Boosting Regression in Business Intelligence which used the principle of Gradient Boosting as foundation and enhanced by incorporation of online learning (streaming data or real-time) which update model incrementally, model adaptation to concept drift, each tree updated multiple times per sample based on a Poisson-distributed count (stochasticity), continuously updates baseline mean and standard deviation (STD) using exponential moving averages (EMA) to normalize residuals and clip outliers, clips residuals based on baseline STD to limit extreme values influencing trees and dynamic model reset per tree on drift detection, reducing complexity and overfitting. Thus, Enhanced Adaptive Gradient Boosting (EAGB) regression is ideal for real-time BI machines.

III. RESEARCH METHODOLOGY

Use Enhanced Adaptive Gradient Boosting Regression Algorithm in Business Intelligence. Ongoing investigation uses quantitative, supervised machine learning approach using principle of Gradient Boosting Regression as the primary predictive modeling technique. Gradient Boosting Regression concept was chosen because of its capacity to model nonlinear relationships, handle mixed feature types, and minimize predictive error through iterative improvement. The design involves data preprocessing, Gradient Boosting model training, tuning its hyperparameter optimization, and evaluating its predictive accuracy on the target problem. The methodology focuses on the development of Enhanced Adaptive Gradient Boosting Regression in Business Intelligence which used gradient boosting as its foundation and integrates online Hoeffding trees, residual clipping, EMA-based baseline updates, and ADWIN drift detection to enable continuous learning, incremental error correction and accurate prediction of continuous values. Dataset for the regression task for this research objective was obtained from secondary sources publicly available datasets. The dataset consists of independent variables (features) $X=\{x_1, \dots, x_n\}$ and a continuous dependent variable (target) y , such as revenue, price, quantity and others. To prepare high-quality input of data for Enhanced Adaptive Gradient Boosting Regression in Business Intelligence, the following procedures were applied: Data cleaning involves handling missing or inconsistent values using techniques like mean, median, or model-based imputation. It also includes removing duplicate records and addressing outliers, either by eliminating them or treating them with robust methods when they carry meaningful information. Feature engineering includes encoding categorical variables using one-hot or ordinal encoding and creating additional meaningful features when needed. A streaming split uses the first W target values as the Warm-Up training

window, and all remaining months enter the Boosting Phase with prequential evaluation (predict then compute error and then train). The prequential Split is a valid method for online learning algorithms such as EAGB. Enhanced Adaptive Gradient Boosting Regression constructs an ensemble of decision trees that learn sequentially from streaming data. The algorithm begins with a Warm-Up window, where the first W target values were used to initialize the baseline mean, variance, and the first tree. After warm-up, all remaining months enter the Boosting Phase, where each new sample is first predicted, then its error is computed, and finally the model is updated a process known as prequential evaluation. Residuals were passed through trees one by one, allowing every tree to correct mistakes of previous trees. Because data arrive in time order, the prequential split is the only valid method for online algorithms such as EAGB, ensuring real-time learning and drift adaptation. The Enhanced Adaptive Gradient Boosting (EAGB) algorithm is an online ensemble-based regression framework designed for real-time business intelligence tasks such as monthly revenue forecasting. EAGB integrates incremental gradient boosting, adaptive Hoeffding regression trees, residual clipping, exponential moving averages, Poisson sampling, and ADWIN drift detection to support non-stationary streaming environments.

Algorithm Steps

Step 0: Input & Initialization:

Input:

Streaming monthly samples data: $\{x_t, y_t\}_{t=1}^T$

N = Number of estimators (trees)

η = Learning rate (Initialize learning rate)

λ = Poisson sampling rate

δ = ADWIN drift threshold

W = Warm-up samples length

C = Residuals Clip multiplier

α = EMA rate

Output:

$\hat{y}_t$ = Predicted revenue for each month t

r_t = Residuals/gradients

Drift resets (per tree)

Performance metrics: (MSE, MAE, R^2 , SMAPE)

CPU/RAM logs

Energy & CO₂ estimates

Model training time

Initialize:

μ = Baseline mean of first W samples

σ = Baseline standard deviation of first W samples

CS = Residual clip scaling multiplier

S = Empty warm-up buffer

Initialize N Hoeffding Trees and ADWIN detectors:

FOR $i = 1, \dots, N$ **DO**

h_i = HoeffdingTreeRegressor()

 ADWIN _{i} = ADWIN(δ)

END FOR

Step 1: WARM-UP PHASE

FOR $t = 1, \dots, W$ **DO**

 Append y_t to warm-up buffer S

$\mu = \text{mean}(S)$

```

 $\sigma = \text{std}(S)$ 
 $r_t = y_t - \mu$ 
Train first tree:
 $h_1.\text{learn}(x_t, r_t)$ 
END FOR

Step 2: BOOSTING PHASE
FOR  $t = W+1, \dots, T$  DO
    (a) Compute Prediction
    # Start with baseline:
     $\hat{y}_t = \mu$ 
    # Add contributions from all trees
    FOR  $i = 1, \dots, N$  DO
         $\hat{y}_t = \hat{y}_t + \eta \cdot h_i.\text{predict}(x_t)$ 
    END FOR
    (b) Compute Residual
     $r = y_t - \hat{y}_t$ 
    (c) Compute Residual Clipping
     $\text{clip\_limit} = \max(1, C \cdot \sigma)$ 
     $r = \text{clip}(r, -\text{clip\_limit}, +\text{clip\_limit})$ 

Step 3: Sequential boosting updates Chain (one residual flow through all trees)
FOR  $i = 1, \dots, N$  DO
    # Sample how many times to train this tree
     $k = \text{Poisson}(\lambda)$ 
    # Train tree  $i$ ,  $k$  times on the current residual
    FOR  $j = 1, \dots, k$  DO
         $h_i.\text{learn}(x_t, r)$ 
    END FOR
    # Predict using current tree
     $p_i = h_i.\text{predict}(x_t)$ 
    # Update residual/gradient for next tree
     $r = r - \eta \cdot p_i$ 
    # Drift Detection for Tree  $i$ 
     $\text{error}_i = |y_t - (\hat{y}_t - r)|$ 
    # Update ADWIN with the new error
    IF  $\text{ADWIN}_i$  detects drift on  $\text{error}_i$  THEN
        Reset  $h_i = \text{new HoeffdingTreeRegressor}()$ 
        Reset  $\text{ADWIN}_i = \text{new ADWIN}(\delta)$ 
    END IF
END FOR # End of all trees

Step 4: Update Baseline Mean and Std (EMA)
 $\mu_{\text{old}} = \mu$ 
 $\mu_{\text{new}} = (1 - \alpha) \cdot \mu_{\text{old}} + \alpha \cdot y_t$ 
 $\sigma^2_{\text{new}} = (1 - \alpha) \cdot \sigma^2_{\text{old}} + \alpha \cdot (y_t - \mu_{\text{old}})^2$ 

Step 5: Output predictions  $\hat{y}_t$ 
The Actual vs Predicted Revenue
Model performance metrics (MSE, MAE,  $R^2$ , SMAPE)
Log resources CPU, RAM usage,
Energy and CO2 emission
Training time
END FOR # End of all streaming samples.

```

Fig. 1 Algorithm of Enhanced Adaptive Gradient Boosting Regression in Business Intelligence

Here we have deliberated the Enhanced Adaptive Gradient Boosting (EAGB) regression Algorithm in Business Intelligence as shown in Fig1. In the Input and Initialization of streaming data: $\{x_t, y_t\}_{t=1}^T$ represents input features x_t and target revenue y_t for each month t . Hyperparameters in the Enhanced Adaptive Gradient Boosting (EAGB) regression algorithm in Business Intelligence were used to control the learning accuracy, stability, and adaptability of the model. In step 1, the Warm-Up Phase ($t = 1, \dots, W$), the algorithm first collects the initial target values y_t and appends them to a warm-up buffer S , which serves as the foundation for computing baseline statistics. Using this buffer, the algorithm calculates the mean $\mu = \text{mean}(S)$ and standard deviation $\sigma = \text{std}(S)$ to represent the initial data distribution. The initial residuals were then computed as $r_t = y_t - \mu$, capturing the deviation of each sample from the baseline. These residuals, along with the corresponding features x_t were used to train the first Hoeffding tree h_1 , establishing the first weak learner of the ensemble. This phase ensures that the model starts with a reliable baseline and prepares the initial tree to incrementally correct future prediction errors during the boosting phase. In step 2, the Boosting Phase ($t = W+1 \dots T$), the model begins by generating predictions starting from the baseline $\hat{y}_t = \mu$ and incrementally adding the contributions of all N Hoeffding trees in the ensemble, scaled by the learning rate η , to form the ensemble prediction. After computing the prediction, the residual $r = y_t - \hat{y}_t$ was calculated to capture difference among actual target and predicted value. To ensure stability and prevent extreme values from disproportionately affecting the model, the residual is clipped using a limit defined as $\text{clip_limit} = \max(1, C \cdot \sigma)$, resulting in a bounded residual $r = \text{clip}(r, -\text{clip_limit}, +\text{clip_limit})$ that will be used for sequential boosting updates in the next steps. In step 3, the Sequential Boosting Updates Phase, the current residual flows through ensemble 1 tree at time, allowing every Hoeffding tree to correct remaining error from previous trees. For each tree $i=1, \dots, N$, the algorithm samples a Poisson-distributed value $k \sim \text{Poisson}(\lambda)$ to determine how many times the tree should be trained on the current residual, introducing stochasticity into the learning process. After updating tree h_i , the tree produces a prediction $p_i = h_i.\text{predict}(x_t)$, which is then used to update the residual for the next tree as $r \leftarrow r - \eta \cdot p_i$, progressively reducing the gradient across the boosting chain. To ensure adaptability in streaming environments, the model computes a drift error $\text{error}_i = |y_t - (\hat{y}_t - r)|$ and passes it to the ADWIN drift detector associated with each tree. If ADWIN signals a drift, the algorithm resets both the tree h_i and its detector, allowing the model to quickly adapt to new data patterns before continuing to the next tree. In step 4, Update Baseline Mean and Standard Deviation, the algorithm adapts its baseline statistics using an Exponential Moving Average (EMA) to account for gradual changes in the data stream. The baseline mean is updated as $\mu \leftarrow (1 - \alpha) \cdot \mu + \alpha \cdot y_t$, allowing recent observations to influence the baseline while still retaining historical trends. Similarly, the variance is updated through $\sigma^2 \leftarrow (1 - \alpha) \cdot \sigma^2 + \alpha \cdot (y_t - \mu)^2$, ensuring that the model continuously tracks changes in data dispersion. This EMA-based update enables the model to remain responsive to slow shifts in the revenue distribution, improving stability and adaptability throughout the boosting process. In step 5, the algorithm outputs the final predicted revenue $\hat{y}_t$ for the current time step and compares it with the actual observed revenue to produced an Actual vs. Predicted evaluation. Log resources consumption like CPU and memory usage over time into a file was recorded for analyze performance. Using this information, model computes performance metrics, namely MSE, MAE, R^2 , SMAPE to assess forecasting accuracy in

real time. Additionally, the system logs resource usage—including CPU load, RAM consumption, energy usage, and estimated CO₂ emissions—to monitor computational efficiency. The training time for each update is also recorded, providing a complete operational and performance profile for every streaming iteration. The Enhanced Adaptive Gradient Boosting (EAGB) regression algorithm provides a robust, scalable, and highly adaptive framework for Business Intelligence applications operating in streaming environments. By combining online Hoeffding trees, sequential gradient boosting, residual clipping, EMA-based baseline updates, and ADWIN drift detection, the model is capable of learning continuously, correcting prediction errors incrementally, and adapting rapidly to changes in revenue patterns. Its design ensures stability during fluctuations, resilience against abrupt drifts, and efficiency in real-time processing. Additionally, the integration of performance metrics, resource monitoring, and CO₂ estimation provides a comprehensive operational perspective, allowing organizations to evaluate both predictive accuracy and computational sustainability. Overall, EAGB regression in BI delivers an effective solution for dynamic revenue forecasting and real-time business decision support in evolving data streams.

IV. FINDINGS

The graph as shown in Fig. 2 that Business Intelligence model used Enhanced Adaptive Gradient Boosting Regression was able to forecast the trends for revenue and able to adjust to revenue changes over time. Model drift detection ensures the predictions remain reliable even when unexpected changes occur. The visual alignment between actual and predicted revenue indicates strong predictive performance.

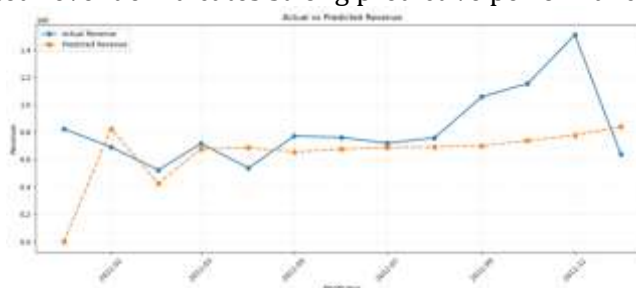


Fig. 2 Time-series line graph comparing actual and predicted revenue.

The Actual vs Predicted Revenue graph demonstrated that the model effectively captures both the trend and seasonality of online retail sales. The predicted revenue closely follows the actual revenue over time, with minor deviations during abrupt spikes or drops. The model dynamically adapts to changes in the data, supported by drift detection, which resets individual estimators when sudden shifts occur. This ensures accurate real-time forecasting and highlights the model's capability to learn from temporal patterns, lag features, and rolling statistics without the need for full retraining. Overall, the visualization confirms that the model provides reliable and responsive revenue predictions suitable for real-time business intelligence applications. Table 1 as shown that Business Intelligence model used Enhanced Adaptive Gradient Boosting Regression the performance, usage of computer resources, energy used, release of carbon dioxide and model training time.

```

===== (a) PERFORMANCE METRICS =====
MSE: 125,776,742,913.58
MAE: 250,195.61
R²: -0.83393
SMAPE: 37.392384%

===== (b) CPU USAGE =====
CPU Usage Before Training: 8.60%
CPU Usage After Training: 40.10%
Average CPU Usage During Training: 29.45%

===== (c) RAM USAGE =====
RAM Usage Before Training: 85.70%
RAM Usage After Training: 86.70%
Average RAM Usage During Training: 85.95%

===== (d) ENERGY & CO₂ EMISSION =====
Estimated Power Usage: 15.9009 W
Energy Usage (estimated): 0.0000062006 kWh
Estimated CO₂ Emissions: 0.000029453 kg CO₂

===== (e) MODEL TRAINING =====
Average Sample Processing Time: 0.005036 s
Total Training Time: 1.4038 s
Model Drift Reset Count: 0

```

Table 1. Performance, usage of computer resources, energy consumption, release of carbon dioxide and model training time.

(a) Performance metrics of the Business Intelligence model used Enhanced Adaptive Gradient Boosting Regression for predicting revenue: (i) MSE (Mean Squared Error): Average of squared differences between actual and predicted values. MSE value was 125,776,742,913.58, a very high value due to squaring large revenue differences. This indicates model's predictions deviate significantly from actual revenue in absolute terms. (ii) MAE (Mean Absolute Error): Average of absolute differences between actual and predicted values. MAE value was 250,195.61. On average, model's predicted revenue differs from actual revenue by 250,196.61 per month. This gives a more 'realistic' sense of error magnitude. (iii) R^2 (R-squared): Proportion of variance in dependent variable (Revenue) explained by model. R^2 value was -0.83393. $R^2 = 1$ indicate perfect prediction, $R^2 = 0$ indicate model performs no better than predicting mean and $R^2 < 0$ indicate model performs worse compared to predicting mean. Negative R^2 (-0.83393) means the model performs worse compared to simply predicting average revenue. This indicates that model is not capturing underlying patterns well, possibly due to insufficient features, highly volatile revenue, or small dataset. (iv) SMAPE (Symmetric Mean Absolute Percentage Error): Average percentage error between actual and predicted values, scaled symmetrically. SMAPE value was 37.392384%. On average, model's predictions were off by 37.392384% which indicates a moderate prediction error in relative terms. (b) CPU usage: (i) CPU usage before training: 8.60% this is the baseline usage of the system before any training started. It indicates the CPU was lightly loaded. (ii) CPU usage after training: 40.10% after the training process finished, the CPU usage increased, which may indicate some residual background processes or that the system hadn't fully returned to idle. (iii) Average CPU usage during training: 29.45%. (c) RAM usage: (i). RAM usage before training: 85.70% the system was already using a large portion of RAM before training started. (ii). RAM usage after training: 86.70% after training, RAM usage increased only slightly. This small change indicates that: the training process didn't allocate significant additional RAM, or the OS released memory quickly after training ended, keeping usage stable. (iii). Average RAM usage during training: 85.95% the average RAM usage during training is almost identical to the "after training" reading, meaning. (d) Energy and CO₂ emission: (i) Estimated Power Usage

was 15.9009 W, which was the average power drawn by the CPU during training. (ii) Energy Usage (estimated) was 0.0000062006 kWh, this was the total energy consumed for the training session. (iii) Estimated CO₂ Emissions was 0.0000029453 kg CO₂. (e) Training time of the model: (i). Average Sample Processing Time: 0.005036 s. This means on average, the model took 0.005036 seconds (≈ 12.8 ms) to process each training sample. This is fairly fast — depending on the model size, it likely indicates: a small/lightweight model, or hardware that is easily handling the workload. (ii). Total training time: 1.4038 s. This means the entire training process was completed in 1.64 seconds. Given the average processing time, this suggests: The dataset is small. (iii) Model drift reset count: 0. this indicates no drift resets were triggered during training. A drift reset usually happens when a model performance becomes unstable, learning rate issues cause divergence, gradients explode, or catastrophic forgetting is detected.

V. DISCUSSION

Business Intelligence model used Enhanced Adaptive Gradient Boosting Regression to solve real-world problems and to determine its performance like accuracy, consumption of computer resources like main memory, CPU, time of training the model, energy consumption and released carbon dioxide. In the future we need to develop other machine learning models to compare which models are better based on the need for the parameters. Overall Implications of the Business Intelligence model used Enhanced Adaptive Gradient Boosting Regression for real-time revenue prediction were (a) Implications of model performance metrics: The current model has shown high errors and a negative R^2 , indicating poor predictive performance. This suggests the model struggles to capture monthly revenue patterns, likely due to small data size and feature limitations. Enhancing feature engineering, model tuning, or using richer datasets could improve forecasting accuracy. (b) Implications of CPU Usage: Overall, CPU usage analysis has shown that the model training was moderately demanding but efficient, ensuring real-time prediction capability while maintaining resource availability for other concurrent tasks. (c) Implications of RAM Usage: Overall, RAM usage analysis has shown that the training process introduces minimal additional memory demand, even in a system that was already operating with high baseline RAM utilization. This consistency indicates that the model was highly memory-efficient, making it suitable for deployment on devices or environments with limited RAM capacity, such as mid-range laptops or systems. The model's low and stable RAM impact ensures that it can support real-time revenue prediction without causing memory-related slowdowns, crashes, or resource conflicts with other business intelligence applications. This makes the model well-suited for real-time BI pipelines where memory efficiency and system stability are essential. (d) Implications of Energy & CO₂: Overall, the energy and CO₂ emission analysis had shown that the model was extremely efficient and environmentally sustainable, consuming only micro-levels of energy during training. The combination of low power usage, minimal total energy consumption, and near-zero CO₂ emissions indicates that the model is ideal for continuous, real-time learning scenarios where predictions must update frequently without raising operational energy costs. This efficiency makes the model especially relevant for modern BI systems that emphasize green analytics, enabling businesses to scale predictive intelligence without significantly increasing their environmental impact. (e) Implication of Training time: Overall, the training time analysis showed that the model was highly efficient, responsive, and stable during training. With extremely fast sample processing and a total training duration of only 1.4 seconds, the model demonstrates clear suitability for real-time revenue prediction in business intelligence settings. The absence of drift resets further underscores its

robustness, suggesting that the model handled the data stream smoothly without encountering instability or concept drift. These characteristics position the model as an excellent fit for continuous, low-latency BI applications where timely adaptation and computational efficiency were essential.

VI. CONCLUSION

This research showed the potential of Business Intelligence model which used Enhanced Adaptive Gradient Boosting (EAGB) Regression to solve real-world business problems in online retail. We began this research by finding the problems, and search for solutions and then implemented solutions for the problems. Business problems can be solved using business intelligence model which used Enhanced Adaptive Gradient Boosting Regression. EAGB regression refer to the real-time, streaming version of Gradient Boosting that updates weak learners' one sample at a time and adapts automatically to concept drift through continuous error monitoring and dynamic model adjustments. It was found that the Green Business Intelligence model which used EAGB Regression was able to adjust to revenue changes over time and able to forecast the trends for revenue in online retail. Model drift detection ensures the predictions remain reliable even when unexpected changes occur. The novelty of this paper was that Conceptual Framework of Enhanced Adaptive Gradient Boosting Regression in Business Intelligence was developed, determine in terms of overall performance, computational resource usage, energy consumption, carbon dioxide emissions, and model training time, and ability to evaluates the trade-off between operational efficiency such as performance, resource utilization, and training time and environmental sustainability including energy consumption and carbon dioxide emissions. The findings of this study highlight important conceptual relationships between Business Intelligence (BI), EAGB Regression which is one of ML techniques, Green Computing (GC) approaches which used to reduce release of carbon footprint from the computer system. The results indicate that ML significantly strengthens the analytical capacity of BI by enabling predictive insights, shifting BI from traditional descriptive reporting toward more intelligent, automated decision-making. Additionally, the study suggests that Machine Learning techniques in Business Intelligence play a crucial role in supporting Green Computing, as BI systems enable organizations to track, analyze, and visualize sustainability metrics such as energy consumption and carbon emissions. This positions BI as a key decision-support mechanism for environmentally responsible IT practices. The relationship between ML and GC further reveals that ML algorithms can optimize energy usage, enhance resource utilization, and reduce operational inefficiencies within computing environments, thereby contributing directly to greener IT operations. When integrated, BI and ML jointly facilitate data-driven sustainability strategies, while GC provides the environmental context within which these technologies operate. Together, these relationships form a conceptual framework where ML provides intelligence, BI delivers actionable insights, and GC ensures that technological advancements align with sustainability goals. We have analyzed Business Intelligence model which used Enhanced Adaptive Gradient Boosting Regression to determine the MSE, MAE, R^2 , and SMAPE, estimated CPU, RAM usage, training time and estimated emissions of CO₂. This study was important for further analysis of Business Intelligence model in terms of accuracy of prediction, energy consumption, estimated CPU, RAM usage, training time and emissions of CO₂.

REFERENCES

- [1] Kopelo Letou and S. Thiyagarajan., "Machine Learning in Business Intelligence", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Vol. 11, No. 11, Pg. 1562–1576, Nov. 2023.
- [2] Davis Gary Alan and Woratschek Charles R., "Evaluating Business Intelligence/ Business Analytics Software for Use in the Information Systems Curriculum," Information Systems Education Journal (ISEDJ), ISSN: 1545-679X, Vol. 13, No. 1, Pg. 23–29, Jan. 2015.
- [3] Gilad Benjamin., Gilad Tamar, "SMR Forum: Business Intelligence-The Quiet Revolution," Sloan Management Review, Vol. 27 No. 4, Pg. 53–61, 1986.
- [4] Friedman Jerome, Hastie Trevor and Tibshirani Robert., "Additive logistic regression: A statistical view of boosting", The Annals of Statistics, Vol. 28, No. 2, Pg. 337–374, 2000.
- [5] Friedman Jerome H., "Greedy Function Approximation: A Gradient Boosting Machine", The Annals of Statistics, Vol. 29, No. 5, Pg. 1189–1232, 2001.
- [6] Friedman Jerome H., "Stochastic Gradient Boosting", Computational Statistics & Data Analysis, Vol. 38, No. 4, Pg. 367–378, 2002.
- [7] Biau G., Cadre B., and Rouvière L., "Accelerated gradient boosting", Machine Learning Vol. 108, Pg. 971–992, 2019.
- [8] Hosen Saikat Md and Amin Ruhul., "Significant of Gradient Boosting Algorithm in Data Management System", Engineering International, ISSN 2409-3629 Vol. 9, No. 2 Pg. 85–100, 2021.
- [9] Mayr Andreas, Binder Harald, Gefeller Olaf and Schmid Matthias., "The Evolution of Boosting Algorithms from Machine Learning to Statistical Modelling", Methods Inf Med; Vol. 53 No. 6, Pg. 419–427, Nov. 2014.
- [10] Bahad Pritika and Saxena Preeti., "Study of AdaBoost and Gradient Boosting Algorithms for Predictive Analytics", International Conference on Intelligent Computing and Smart Communication, Conference paper, Pg. 235–244, Dec. 2019.
- [11] Chen Sihang, Zhang Na, Shui Biyuan, Zhang Tiantian, Li Chunman and Gong Qingjun., "Real-Time Prediction of Overpressure-Induced Pipeline Failure via Edge-Optimized Machine Learning", Energy Science & Engineering, Vol. 13, Pg. 4730–4744, 2025.
- [12] Nallakuruppan M.K., Varun C. Sai, Dhanaraj Rajesh Kumar, Tiwari Shashi Kant, Malathi V., Pamucar Dragan and Delen Dursun., "Reliable generative interpretable framework for efficient predictive analysis of air quality index", Egyptian Informatics, Vol. 100773 Pg. 1–18, 2025.
- [13] Rafiq Islam, "Ensemble Methods: Gradient Boosting - A detailed overview", Pg. 1–5, Oct. 2024.
- [14] Natekin Alexey and Knoll Alois KnollAlois, "Gradient boosting machines, a tutorial", Front. Neurobot., Vol. 7, Pg. 1–21, Dec. 2013.
- [15] Kopelo Letou and S. Thiyagarajan., "Green Computing of Machine Learning in Business Intelligence", International Journal of Environmental Sciences, ISSN: 2229-7359 Vol. 11 No. 15, Pg. 2418–2434, 2025.
- [16] Mahesh Batta, "Machine Learning Algorithms - A Review", International Journal of Science and Research (IJSR) ISSN: 2319-7064, Vol. No. 9 Issue No. 1, Pg. 318–386, Jan. 2018.
- [17] Pega Documentation, "Adaptive Gradient Boosting overview."
<https://docs.pega.com/bundle/platform/page/platform/decision-management/adaptive-boosting-algorithm.html>, Oct. 2025.