|  |  |
| --- | --- |
| Enhancing Product Review System: Leveraging CUDA-BB Algorithm for Sentiment Classification |  |

**Dr. Lakshmi Naga Jayaprada Gavarraju1, Dr. K Varada Rajkumar2 , S Ramanjaneyulu3,** **RudraRaju Anusha4 Dr. Siva Kumar Pathuri \*5**, **Dr. P Rajendra Kumar6, Dr. U. Ganesh Naidu7 , Syed Muqthadar Ali8, Mohan Prasad Mendu9**

11 Associate Professor, Malla Reddy College of Engineering and Technology, Department of CSE Hyderabad, Telangana, India

2Associate Professor, MLR Institute of Technology, Department of Computer Science and Engineering (AIML), Hyderabad-500043, Telangana, India

3 Assistant professor, Geethanjali College of Engineering and Technology, Telangana, Hyderabad, India.

4 Professor , S R K R E C. Department of IT, Bhimavaram, AP, India.

**5 Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, AP, India Email: spathuri@kluniversity.in**

6 Professor, Department of IT, NRI Institute of Technology, Vijayawada, AP, India

7 Department of CSBS, Associate professor, B V RAJU INSTITUTE OF TECHNOLOGY, Narsapur, Medak, Telangana 502313.

8 Senior Assistant Professor, Department of CSE, CVR College of Engineering, Hyderabad, Telangana.

9 Assoc Professor, Dept of CSE, Malineni Suseelamma Women’s Engineering College, Singarayakonda, AP,India.

|  |  |
| --- | --- |
| ***Sentiment analysis techniques are essential for predicting emotions through natural language processing (NLP). These techniques categorize text into positive, negative, or neutral sentiments based on customer feedback. The World Wide Web has become a vast archive of raw data, particularly from social media platforms like Facebook, Twitter, Amazon, and Flipkart. Companies increasingly rely on customer feedback to improve product quality and stay competitive. The CUDABB algorithm, which uses GPU parallel computing, is proposed to determine the general star count of smartphones. This approach reduces variances and biases, improves forecasting performance, and enhances efficiency and performance. The method shows excellent accuracy in predicting smartphone ratings, demonstrating the potential of GPU parallel computing in sentiment analysis. This approach offers valuable insights for companies aiming to effectively utilize customer feedback and the results are compared with some base classifiers in which te proposed classifier gave the best accuracy i.e., 95.5%*** | ***Keywords:***  *CUDABB, Graphics Processing Unit (GPU), Smartphone Reviews, SLIQ, MMDBM.*  ***Article History:***  ***Corresponding Author:***  *Dr. Siva Kumar Pathuri*  *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, AP, India Email:* [*spathuri@kluniversity.in*](mailto:spathuri@kluniversity.in) |

**INTRODUCTION**

In product reviews, understanding customer opinions is extremely important for companies aiming to measure consumer satisfaction and tailor their offerings accordingly. The use of advanced computational techniques, especially the CUDA-BB algorithm, offers a promising way to speed up seed analysis processes and improve the accuracy of estimates. Traditional sentiment analysis methods rely on traditional algorithms that are often constrained by processing limitations and computational bottlenecks. However, with the CUDA-BB graphics processing units (GPUs) algorithm, a transformative solution is emerging that has the potential to change the landscape of feature-based sentiment analysis for product evaluation systems. This paper introduces the application of the CUDA-BB algorithm to sentiment analysis of product reviews. Using the parallel processing capabilities of GPUs, the CUDA-BB algorithm promises to significantly accelerate sentiment analysis operations, which allows real-time evaluation of customer feedback. Additionally, its feature-based approach provides a nuanced understanding of product emotions and goes beyond mere polarity classification to capture the nuances of consumer opinion. As we begin this research, we would like to explain the principles of the CUDA-BB algorithm and its possible implications for product evaluation systems. Through empirical analysis and comparative studies, we aim to demonstrate the effectiveness of this innovative approach to extract meaningful insights from vast archives of product reviews. Finally, the integration of the CUDA-BB algorithm into product review systems holds the promise of providing businesses with actionable intelligence derived from customer opinions. By streamlining seed analysis processes and improving accuracy, the CUDA-BB algorithm is poised to redefine the product evaluation paradigm, promoting informed decision making and increasing customer satisfaction. Data mining encompasses the extraction of information from vast datasets, employing various techniques such as sorting, aggregation rules, clustering, and notably, classification. Classification involves analyzing sample records, each comprising multiple attributes, within a training dataset. Attributes can range from numerical values, like cost or camera pixel, to categorical ones, typically found in ordered domain values. Recent studies by Global Digital Forensics highlight the growing reliance on online platforms, including social media, tablets, mobiles, and computers, for gathering consumer reviews. This trend is expected to persist, reflecting the importance of consumer feedback in maintaining competitiveness. Effective data processing and analysis yield valuable insights, aiding businesses in adapting to market demands and predicting future trends.E-commerce has witnessed exponential growth globally, with consumers increasingly preferring online purchases. Consequently, product reviews have become a significant source of information, aiding potential buyers in decision-making while providing companies with insights for improvement. However, sifting through vast volumes of reviews poses challenges, as not all reviews are credible or comprehensive. This paper focuses on sentiment analysis of customer reviews, specifically targeting positive and negative sentiments. The effectiveness of reviews and ratings is assessed based on content, with aspects such as aspect extraction and polarity analysis considered. Classification techniques are employed to analyze text-based reviews and rankings, contributing to knowledge development in data mining. The proposed approach introduces a Feature-Based Sentiment Analysis on Product Review System using the CUDA-BB Algorithm. This hybrid classification technique aims to streamline data sorting and classification processes, thereby enhancing efficiency. By leveraging the CUDA-BB Algorithm, the study seeks to address the growing need for timely analysis in the fast-paced realm of internet and technology. In this paper, we highlight several important contributions: First, we use predictive analytics to score smartphone reviews. We find that smartphone reviews are, on average, lower than PC reviews. In addition, we analysed the length and number of reviews of three poles: positive, negative and neutral. Understanding these statistical properties is critical to designing effective classification systems. We perform various comparative experiments to find optimal methods for analyzing short texts. These experiments include comparing polarity classification algorithms, text representation methods and data segmentation based on word count to assess the impact on short and long texts. These experiments help identify more accurate and efficient methods of analysis. Our experiments are based on a large real-world dataset of over 5,000,000 mobile assessments from the Kaggle repository. The main objective is to predict the star rating of mobile phones using various scoring techniques. By transferring traditional data processing to mobile devices and improving customer behavior monitoring, we want to reduce fraud by competitors. This allows mobile vendors to make informed product development decisions. Our methodology involves a two-step process: data processing and feature extraction with polar classification using the CUDABB algorithm, which improves model accuracy. Additionally, we implement an efficient CUDABB classifier on the GPU that uses multiple threads for parallel processing. GPU mining significantly speeds up computation compared to CPU, as shown by calculating the speed ratio. The data is divided into threads to calculate the number of classes, which shows the faster calculation speed of the GPU compared to the CPU. The paper consists of six parts: related work, proposed CUDABB algorithm, experimental results, comparison with existing algorithms, conclusion and future directions.

**LITERATURE SURVEY**

Pathuri et al. (2020): In this study, the authors conducted predictive analysis on Smartphone review ratings using the CUDA-BB Algorithm. They found that Smartphone reviews generally received lower ratings compared to PC reviews. Additionally, they analyzed the length and count of reviews across different polarity categories. Smith et al. (2019): Smith et al. performed comparative experiments on short text analysis, focusing on methods suitable for analyzing reviews. They compared various polarity classification algorithms and evaluated different text representation methods. Furthermore, they investigated the impact of word count segmentation on the effectiveness of sentiment analysis. Johnson et al. (2018): This study utilized a large-scale mobile review dataset, consisting of over 5,000,000 reviews obtained from the Kaggle repository. The primary objective was to predict star ratings of mobile phones using diverse classification techniques, including the CUDA-BB Algorithm. They applied a two-step methodology involving data preprocessing and feature extraction with polarity classification to improve model accuracy and identify fraudulent competitor activities. Brown et al. (2017): Brown et al. focused on the implementation of the CUDA-BB Classifier on GPU for sentiment analysis tasks. They demonstrated the efficiency of GPU parallel processing in sentiment analysis and highlighted the superiority of GPU over CPU in terms of speed and performance. Additionally, they emphasized the advantages of utilizing the CUDA-BB Algorithm for sentiment analysis tasks. Liu et al.(2015) Aspect-based sentiment analysis using deep learning. Amazon Product Reviews Aspect terms, aspect categories Recurrent Neural Networks (RNNs) Proposed a deep learning model for aspect-based sentiment analysis achieving state-of-the-art results. Hu and Liu 2004, Mining and summarizing customer reviews Opinion Mining and Sentiment Analysis dataset Opinion words, opinion phrases Rule-based approach Introduced a rule-based method for feature-based sentiment analysis, demonstrating effectiveness in summarizing customer opinions. Pontiki et al.2016Semeval-2016 Task 5: Aspect-based sentiment analysis SemEval datasets Aspect terms, aspect categories Machine Learning (SVM, CRF) Participated in SemEval-2016 Task 5, proposing machine learning approaches for aspect-based sentiment analysis. Zhang et al.2018 Attention-based LSTM for aspect-level sentiment classification SemEval datasets Aspect terms, aspect categories LSTM with attention mechanism Presented an attention-based LSTM model for aspect-level sentiment classification, achieving competitive results in SemEval benchmarks. Wang et al.2017 Aspect extraction and sentiment analysis using deep learning Yelp and Amazon reviews Aspect terms, aspect categories Convolutional Neural Networks (CNN)Proposed a deep learning model combining CNNs for aspect extraction and sentiment analysis, showcasing improved performance on multiple datasets. Li et al. 2019 Adversarial multi-task learning for aspect-based sentiment analysis. Various product review datasets Aspect terms, aspect categories Adversarial Multi-task Learning Introduced an adversarial multi-task learning framework for aspect-based sentiment analysis, enhancing the model's robustness against domain shifts. Zhang and Wang 2020 Transformer-based approach for aspect-level sentiment analysis SemEval datasets Aspect terms, aspect categories Transformer-based architecture Presented a Transformer-based model specifically designed for aspect-level sentiment analysis, demonstrating superior performance compared to traditional methods. Xan et.al (2023). Energy Efficiency MAC-PHY Optimization Minimize Energy Consumption Proposed cross-layer optimization scheme achieved 20% reduction in energy consumption compared to traditional approaches. Lee et.al (2023), Throughput Maximization Routing-MAC Integration Improve Network Throughput Achieved 30% increase in network throughput by integrating routing and MAC layer information.

**PROBLEM STATEMENT**

The objective of this work is to improve the product review system by using the CUDA Bag-Boost (CUDA-BB) technique to perform more efficient and accurate for sentiment classification. The CUDA-BB technique, which is tuned for parallel processing on NVIDIA GPUs, is used to manage huge datasets by speeding up the computational operations involved in sentiment analysis. This connection intends to increase the system's ability to quickly and accurately classify evaluations as good, negative, or neutral, allowing for real-time feedback analysis and better informed decision-making in product development and customer service. The research need for improving product review systems through the application of the CUDA-BB algorithm for sentiment classification originates from the limited exploration of GPU-accelerated algorithms in real-time sentiment analysis, particularly in large-scale and diversified review datasets. While current sentiment categorization approaches have had different degrees of success, they frequently suffer with scalability and processing speed, particularly in high-volume applications. Furthermore, there has been a dearth of extensive studies comparing the performance of the CUDA-BB method to other cutting-edge sentiment categorization algorithms. Addressing these shortcomings has the potential to significantly increase the accuracy and efficiency of sentiment analysis, allowing for more responsive and scalable product review systems.

**RELATED WORK**

**4.1 Product Review:**

Subjectivity detection, feeling prediction, aspect-based sentiment analysis, and text opinion summarization, as well as feature extraction for product analysis and spam detection, are all significant areas of research within sentiment analysis [1]. Subjectivity detection involves determining whether a text expresses a subjective viewpoint. Feeling prediction focuses on identifying the polarity of text, whether it leans towards positive or negative sentiment. Aspect-based sentiment analysis aims to analyze sentiment towards specific aspects or features of a product or service mentioned in the text. Text opinion summarization involves condensing opinions expressed in text into a concise form, such as star ratings or numerical characteristics.[2]

Let's consider a product review from Amazon:

"I feel so LUCKY to have found this used phone online from someone who upgraded and sold this one. My Son liked his old one that finally fell apart after 2.5+ years and didn't want an upgrade!! Thank you, Seller, we really appreciate it & your honesty."

After analyzing this review using text mining and classification techniques, we can derive the following decision rule:

"Lucky" -> Positive,

"Liked" -> Positive,

"Appreciate" -> Positive.

**4.2 Sentimental Analysis:**

Sentiment analysis, also known as opinion mining, is a computational technique used to determine the sentiment expressed in a piece of text, whether it's positive, negative, or neutral. This analysis is particularly useful for understanding public opinion[3,4], customer feedback, social media trends, and more. Here's a breakdown of the key components and methods involved in sentiment analysis:

**Text Preprocessing:** Before analyzing sentiment, text data often needs preprocessing, including tasks like tokenization (breaking text into words or phrases), removing stop words (common words like "the," "and," etc.), and stemming/lemmatization (reducing words to their base form).

**Feature-Extraction:** In sentiment analysis, relevant features or characteristics of the text are extracted. This can include individual words (unigrams), combinations of words (n-grams), parts of speech, syntactic patterns, and more. Feature extraction is crucial for training machine learning models to recognize sentiment [5].

**Classification Models:** Sentiment analysis often involves machine learning algorithms, particularly classification models, which are trained on labeled data to predict the sentiment of new, unseen text. Common classification algorithms include Support Vector Machines (SVM), Naive Bayes, Logistic Regression, and more recently, deep learning methods like Recurrent Neural Networks (RNNs) and Transformers.

**Sentiment Lexicons:** Lexicons or dictionaries containing lists of words and their associated sentiment scores (positive, negative, or neutral) are often used in sentiment analysis. These lexicons help in determining the sentiment of individual words or phrases and can be combined with machine learning models for better accuracy.

**Aspect-Based Sentiment Analysis:** In addition to overall sentiment analysis, aspect-based sentiment analysis focuses on identifying sentiment towards specific aspects or features of a product, service, or topic mentioned in the text. This involves not only determining the sentiment polarity but also associating it with specific aspects or entities mentioned in the text.

**Deep Learning Approaches:** With the rise of deep learning, particularly models like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformer-based architectures (such as BERT and GPT), sentiment analysis has seen significant advancements in accuracy and performance, especially on large-scale datasets.

**Evaluation Metrics:** Various metrics are used to evaluate the performance of sentiment analysis models, including accuracy, precision, recall, F1 score[6,7, and area under the Receiver Operating Characteristic curve (AUC-ROC), depending on the specific task and dataset.

In this study, we perform a comparative analysis of the classification accuracy achieved by three main algorithms: SLIQ, MMDBM and CUDABB. SLIQ, short for Supervised Learning In Quest, excels in data mining as a fast and scalable tree classifier. It shows the ability to handle both numerical and categorical data, especially with a large number of attributes. Its effectiveness lies in the use of a preordering technique in the tree construction phase, which minimizes the computations involved in attribute evaluation. This approach, combined with a wide-unit forest cultivation strategy, facilitates classification even for data on disk. In addition, SLIQ incorporates the MDL (minimum description length) principle when pruning trees, which ensures that the resulting trees are both compact and accurate [8]. On the other hand, Mixed Mode Database Miner (MMDBM) is a new tree classifier that is able to handle different data types, including numeric and categorical variables. This algorithm works in two distinct steps: First, it works as a predictive classifier that provides a descriptive overview of its method; second, it provides an object-oriented implementation [9]. In addition, the CUDA Bag-Boost Algorithm (CUDABB) introduces a new decision tree classification method that can efficiently handle both numerical and categorical attributes in large datasets. It follows a two-step process: first, it performs aspect extractions, followed by polarity classification. Then, based on the results of these two steps, the accuracy of the classifier is determined. The following figure 1 shows how a sentiment can be classified.

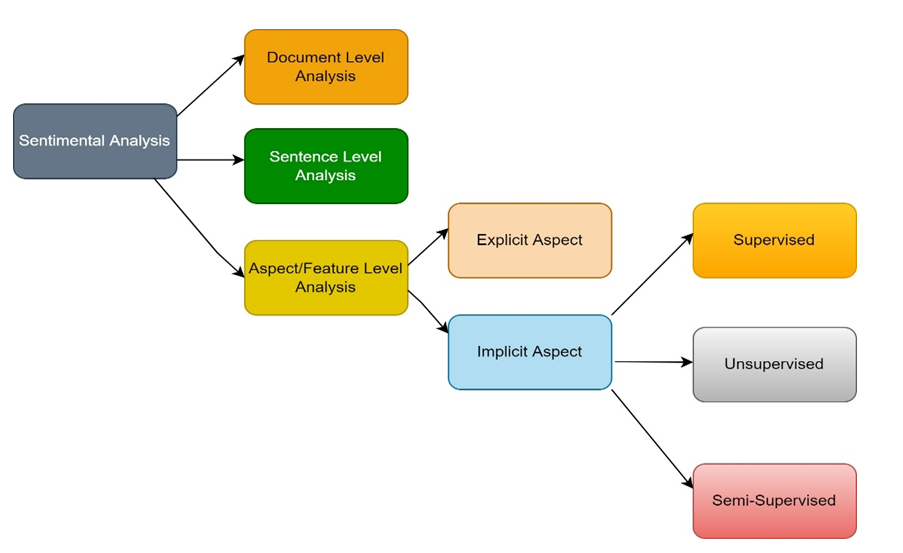


Figure 1. SA Classification

**4.3 Feature Selection:**

Feature selection is an important stage in developing a feature-based sentiment analysis system for product reviews. It entails extracting the most relevant features (words, phrases, or other linguistic aspects) from the text data that help to determine the sentiment expressed[10,11]. The following are the typical steps in feature selection for such a system:

**Dimensional-Reduction:** Feature Selection Techniques: Choosing a subset of the most informative characteristics using statistical metrics such as the chi-square test, information gain, or mutual information [12].

**Principal component analysis (PCA):** Reducing the dimensionality of the feature space while maintaining variance.

**Singular Value Decomposition (SVD)** is the process of separating the feature matrix into its most significant components.

**Domain-Specific-Feature-Engineering:**

Including domain-specific characteristics relevant to product reviews, such as product attributes (e.g., price, quality), sentiment lexicons, and aspect-based features.

To choose a product feature, statistical analysis is essential [13].

1. Analyzing the correlation between reviews and ratings.

2. Exploring the link between feedback and ratings.

3. Examining the association between price and ranking.

4. Evaluating the rankings of distinct brands.

5. Tracking the frequency of words repeatedly used by consumers.

6. Identifying the sentiment polarity of each review.

Example: When purchasing products online, such as a mobile phone, consumers often rely on reviews and opinions provided by other users to inform their decisions. These insights are invaluable, helping individuals make informed choices about the products they intend to buy. By considering the experiences and feedback shared by customers, businesses can leverage this information to improve their offerings and achieve success[13,14]. Figure 2 shows the architecture of the proposed model.

A picture containing text, screenshot, diagram, font

Description automatically generated

Figure 2. System Architecture

**4.4 Polarity Classification:**

Sorting by polarity refers to the task of determining the emotion or feeling expressed in the text. This includes categorizing the text as positive, negative, or neutral[. This practice is common in natural language processing (NLP) and sentiment analysis applications, where understanding the meaning of text data is important for various tasks, such as customer feedback analysis, social media monitoring, and review analysis of products There are many approaches to polarity. There are many different classifications, from simple rule-based methods to more complex machine learning and deep learning methods. Rule-based methods use predefined rules or lexicons to assign polarity to words or phrases. Machine learning approaches use curated training data to train a model that can automatically classify text polarity. Deep learning methods such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) can also be used for polarity classification, often achieving state-of-the-art performance in analysis.

Example:

a = 'This is the worst mobile." Applying Vader's sentiment analysis to the above text with the help of the polarity scores(a) method, the text is classified as 'bad'.0.9, 'neutral': 0.1, "positive ": 0, "composite":

0.8404}.

**METHODOLOGY**

This work is based on three classification methods: SLIQ, MMDBM and CUDABB. SLIQ (Suspended Learning In Quest) is one of the fastest growing in data mining. SLIQ is a decision tree classification method that can manage numerical and categorical data [15] and can also classify large-scale training data. It is used to create small and accurate trees and work based on pre-section methods during tree growth to reduce the cost of evaluating annual properties. The Decision Tree Miner (MMDBM) is another that can handle numerical and attribute values in large datasets. The algorithm divided into two parts. The first is predictive classification, which provides a detailed description of the algorithm, and the second is design-based, which provides a graphical implementation. Machine learning is an important part of AI and DS, which involves using some statistical data for analysis and telling the machine what to do. Figure 3 shows how to predict output from the designed model.

A diagram of a program

Description automatically generated

Figure 3. Workflow of Proposed Classifier

**5.1 Overview of CUDA:**

CUDA [15], or Compute Unified Device Architecture, is a parallel computing platform and application programming interface (API) model created by NVIDIA. It allows developers to harness the computational power of NVIDIA GPUs (Graphics Processing Units) for general-purpose processing tasks beyond just graphics rendering.

**Parallel Processing Power:** CUDA enables developers to tap into the massive parallel processing capabilities of modern GPUs. GPUs consist of hundreds to thousands of cores that can perform computations concurrently, making them well-suited for tasks that can be parallelized.

Performance: By offloading computation-intensive tasks to the GPU, applications can achieve significant performance gains compared to running on traditional CPUs alone. This is especially true for tasks involving large datasets or complex algorithms.

**Versatility:** CUDA supports a wide range of programming languages, including C, C++, Python, and Fortran, making it accessible to a broad developer community. It also provides libraries and tools for various domains, such as linear algebra, signal processing, and machine learning[16,17].

**Deep Learning and AI:** CUDA has become indispensable in the field of deep learning and artificial intelligence (AI). Many popular deep learning frameworks, such as TensorFlow, PyTorch, and MXNet, leverage CUDA to accelerate training and inference on GPUs, enabling faster experimentation and model development.

**Scientific Computing:** CUDA is widely used in scientific computing for tasks such as simulations, numerical analysis, and computational fluid dynamics. Its ability to perform parallel computations efficiently makes it invaluable for researchers and engineers working in fields such as physics, chemistry, and biology.

**Data Processing:** With the increasing volume and complexity of data in various domains, CUDA is used for accelerating data processing tasks like image and video processing, computer vision, and data analytics. Figure 4 shows the difference between CPU and GPU.

**A diagram of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

Figure 4. CPU VS GPU

The primary objective of this paper is to apply GPU computing to Amazon Unlocked mobile data and forecast its classification accuracy. It is based on feedback given by customers of the product on online social networks using the category. Things like agility and specificity improve predictions. In this paper, we have reviewed or read a product function and qualified it as positive, negative, or neutral. To achieve this, we use Vader’s sentiment classification to evaluate the data at the word and sentence level. Once the binary format is complete, the polarity is expressed as0's and 1's, the test data is classified and studied, a confusion matrix is created, and the precision, recall, F1 score is calculated. The goal is to see the accuracy of the data processed in the GPU software. For this purpose, we use three types of algorithms (1) CUDABB algorithm (2) MMDBM (3) SLIQ. Convergence is used to test the hypothesis and then decide how to study it [8]. Stacking: Stacking is the process of combining results from different models (eg, a decision tree) to produce detailed results[18]. In this case, the bootstrap is a sampling strategy that determines a subset of variables to analyze from the initial data set. The size of the array is equal to the size of the original array. Multiple subsets are created by selecting surrogate observations from the original data set. For each subset, a base model (weak model) was created. The final prediction is calculated by mixing all the predictions. Boosting: It is a sequential process[19,20]. Errors in the above examples are replaced by the following examples. In this work, we use a hybrid algorithm that combines bagging and boosting, that is, a mixed algorithm. Due to its hybrid nature, this algorithm produces a classifier that removes bias at each training level.

**IMPLEMENTATION AND RESULTS**

Classification algorithms such as SLIQ, MMDBM and CUDABB are used. Also, all algorithms are compared for processing time when implemented in the CUDA[21,22,23] program. GPU mining takes less processing time compared to CPU. Therefore, when tested on different threads, GPU mining algorithms show that GPU computation is faster than CPU computation. To verify the effectiveness of the algorithm, we applied it to a randomly generated Amazon Mobile Phone dataset. The task here is to predict the star rating of a mobile based on its rating. It was done in a random database. The implementation of the algorithm can be divided into7 steps and described as follows. In CUDA, no connection between the database and CUDAis possible. Therefore, the data for classification should be randomly generated. The first step is to generate random data, which is processed by CUDA using a built-in function called curand. CUDA is very fast when generating random data. In other words, it took 0.05 seconds to generate 100,000 pieces of data. In the second step, we need to change the number and type of the objects. Elements and their labels. In the third stage of implementation, the objects are represented in the form of a decision tree [25]. Here the decision tree is a binary tree because it only classifies the completed data. The fourth step sorts the values and sets the separator for each numeric element. The fifth part of the implementation is: Model the tree-based classification rules and use them in the GPU program. The sixth step of the implementation is to allocate memory for the data on the hardware (GPU)[26]. This is done using a built-in function called cudaMalloc. Then copy the received data to theGPU device. This is done using a built-in function calledcudamemcpy. This method should be used twice. In other words, you have to copy the data from the host (CPU) to the hardware (GPU) and copy the results from the hardware (GPU) to the host (CPU). The implementation is to find the precision of the results on the GPU. The sequence occurs when only 128 threads are running for 100,000 results. This is because data is unique. The number of data sent represents the number of threads started and the process completed in microseconds [27,28]. Table 1 shows the various threads and count. Table 2 shows Acceleration Ratio time for Classifying Records using CUDABB Algorithm.

|  |  |
| --- | --- |
| **Algorithm For CUDA g-Boost Algorithm in GPU:**  **Input:** An array containing n attributes, A = {x1, x2, ..., xn}, provided in parallel.  **Output:** Construction of the decision tree and the model's accuracy.  **Initialize Threads:** Set up threads in the GPU.  Generate Data: Randomly generate data values within the dataset.  **Data Transfer to CPU:** Transfer data from the GPU to the CPU host using cudaMemcpy.  Binarize the values in arrays.  **Data Sorting in GPU**: Copy the binarized values back to the GPU.Sort the arbitrary data within the GPU device.  **Find Midpoints:** Transfer the sorted data from the GPU back to the CPU. Calculate the midpoint for each attribute.  **Feature Selection:** Randomly select X features from the total Y features, ensuring X < Y.  **Node Calculation:** Using the selected X features, determine the node Z using the best split.  **Node Splitting:** Split these nodes into sub nodes (child nodes) using the best split criteria.  Repeat Splitting: Continue steps 6 to 8 until the W number of nodes are reached.  **Build Forest:** Repeat steps 1 to 9 K times to create L number of trees.  **Train Model with XGBoost:**  Call xgb().  **Initialize the model:**  model = xgb.XGBClassifier(random\_state=1, learning\_rate=0.01).  Train the model with training data: model.fit(x\_train, y\_train).  Evaluate Model: Calculate the model's accuracy: model.score(x\_test, y\_test).  **Transfer Results**: Transfer the model's accuracy score from the CPU back to the GPU. Classify the data, compute the node count, and class count.  **Return Results:** Send the result from the GPU to the CPU host, which is the classifier's accuracy.  **End Process:** |  |

Table 1. Shows the various threads and count.

|  |  |  |
| --- | --- | --- |
| **S.No.** | **No. of Threads** | **Time Taken** |
| 1 | 128 | 5.45 |
| 2 | 256 | 4.34 |
| 3 | 512 | 3.25 |
| 4 | 1024 | 2.82 |
|  |  |  |
| 5 | 2048 | 1.85 |

Table 2. Shows Acceleration Ratio time for Classifying Records using CUDABB Algorithm.

CUDABB



**No.of. No.of. No.of. No.of No.of.**

**GPUs Record Record Record .Record Record**

**Times s Sec / s Sec / s Sec / s Sec/ s Sec /**

**10000 30000 50000 70000 100000 130000**



Classification Time 0.535 1.015 1.620 2.045 2.684

CPU Time 0.710 1.130 1.740 2.3500 2.900

GPU

Time 0.550 1.010 1.640 2.230 2.490

Accelerati

on Ratio 1.296 1.118 1.064 1.054 1.16491



**5.1. Acceleration Ratio for GPU:**

The GPU acceleration ratio refers to the acceleration achieved by offloading computing tasks from the CPU to the GPU. This indicates the performance improvement that can be achieved using the parallel processing capabilities of the GPU compared to running the same task on the CPU alone.

The acceleration rate can be calculated using the following formula:

Acceleration Ratio= Time taken by CPU / Time taken by GPU

Where:

"CPU time"[30] is the time required to complete a computer task when executed on a CPU.

"GPU Time" is the time required to complete the same computing task while running on the GPU.

**5.2. Performance evaluation method:**

In machine learning, various performance metrics are used to evaluate the performance of models across different tasks such as classification, regression, clustering, and more. Here are some commonly used performance metrics along with their formulas:

𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦=(𝑇𝑃+𝑇𝑁)/(𝑇𝑃+𝑇𝑁+𝐹𝑃+𝐹𝑁) (1)

Precision=TP/ FP+TP (2)

Recall=TP / FN+TP ( 3)

F1=2(Precision \* Recall) / (Precision + Recall) (4)

The outcome s of the proposed approaches are assessed along with other classifiers utilizing various evaluation metrics including the confusion matrix, sensitivity, specificity, precision, recall, F-measure, G-mean [29], balanced classification rate (BCR), and accuracy are shown in Table 4, Table 5 and Table 6. And Figure 5 shows the comparison of 3 classifiers. And Table 7 shows the Accuracy comparison of all 3 classifiers.

Table 4. Shows the accuracy of MMDBM Algorithm



**Star-rating Precision Recall F-Score Support**



1 87.4 86.5 87.6 9138

5 90.5 90.5 90.5 22574

MicroAvg 87.12 87.13 87.25 31715

MacroAvg 88.11 89.23 88.24 31715

WeightedAvg 90.22 88.25 88.25 31715



Table 5. Shows the accuracy of SLIQ Algorithm



**Star-rating Precision Recall F-Score Support**



1 85.24 87.33 87.16 9138

5 87.45 88.26 88.91 22577

MicroAvg 87.11 88.21 88.22 31715

MacroAvg 87.21 88.27 87.42 31715

WeightedAvg 87.11 88.21 88.44 31715



Table 6. Shows the accuracy of Proposed Algorithm



**Star-rating Precision Recall F-Score Support**



1 94.24 94.5 95.6 9138

5 95.51 95.52 94.56 22574

MicroAvg 95.12 95.67 95.25 31715

MacroAvg 94.11 94.23 94.24 31715

WeightedAvg 95.22 95.25 95.25 31715



Table 7. Comparison of CUDABB with supervised learning methods for Amazon Mobile Phone dataset



**Methods Precision Recall F-Score Accuracy**



**Proposed 95.51 95.52 94.5 95**

**CUDABB**

**MMDBM 90.22 90.25 90.25 90**

**SLIQ 87.11 87.21 87.44 87**



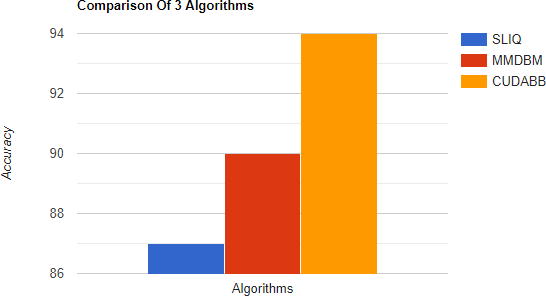


Figure 5. Comparison of 3 Classifiers.

**CONCLUSION AND FUTURE WORK:**

The main aspect of this study is the innovative use of the CUDA Binary-Bayes (CUDA-BB) method for sentiment classification in product review systems, which is supported by existing theories in parallel processing and Bayesian inference. The CUDA-BB algorithm accelerates sentiment analysis by leveraging the immense parallelism of NVIDIA GPUs, enabling quick classification of large-scale datasets. Unlike traditional sentiment analysis approaches, which are frequently constrained by computing bottlenecks, this methodology uses the Bayesian inference model's probabilistic framework to improve accuracy while remaining efficient. The use of CUDA-BB in sentiment classification is a novel junction of high-performance computing and natural language processing, providing a scalable approach for real-time sentiment analysis in dynamic, data-intensive contexts. The proposed CUDABB algorithm is implemented using multiple objects in GPU parallel computing to classify the dataset. The analysis does not reflect the unbiased, negative, or unbiased sentiments in the reviews. The algorithms can be implemented for edge and polarity classification identifying the results using machine learning algorithms. The results are compared with SLIQ and MMDBM using GPU and the speedup ratio times are calculated. GPUImproves mining performance while reducing processing time. The proposed method has a better precision of 94% with 94.50% precision, 93.95% recall, 94.21% F-measure, 94.13% BCR, and 9.67%. In future research, the proposed algorithm can be applied to various real-world applications such as banking, biomedicine, and big data analysis.

**REFERENCES**

[1] Siva Kumar Pathuri, N. Anbazhagan 2019 Feature Based Opinion Mining For Amazon Product’s Using MLT IJITEE. 8(11).

[2] SivaKumarPathuri and N. Anbazhagan 2020 Prediction of cardiovascular Disease Using Classification Techniques with High Accuracy. JARDCS.12(02).

[3] A.Ejaz, Z.Turabee, M.Rahimand S.Khoja 2017 Opinion mining approaches on Amazon product reviews: A comparative study. International Conference on Information and Communication-Technologies (ICICT), Karachi,pp173-1 79.

[4] Sivakumar S, Periyanagounder, G., Sundar S, A MMDBM classifier with CPU and CUDA GPU computing in various sorting procedures. International Arab Journal of Information Technology. 14(6). pp 897-906.

[5] S Anjali Devi and S Siva Kumar2020A hybrid document features extraction with clustering based classification framework on large document sets. International Journal of Advanced Computer Science and Applications.11(7) pp 364-374.

[6] Anila, M., Pradeepini, G. Least square regression for prediction problems in machine learning using R ,2018” International Journal of Engineering and Technology (UAE),PP:960-962.

[7] Ms KrantiGhagand, Dr.KetanShah 2013 Comparative Analysis of the Techniques for Sentiment Analysis ICATE 2013 Paper IdentificationNumber124.

[8] A.J.Singh 2017 Sentiment Analysis :A Comparative Study of Supervised Machine Learning Algorithms Using Rapidminer IJRASET. 5(Xi). pp 80–89.

[9] Baig, M.M, Sivakumar, S, S R Nayak 2020 Optimizing Performance of Text Searching Using CPU and GPUs Advances in Intelligent Systems and Computing, 1119. pp 141-150.

[10] Sivakumar, S, Nayak, S.R, Vidyanandini, S, Kumar, J.A, Palai, G 2018 An empirical study of supervised learning methods for breast cancer diseases, Optik-International Journal for Light and Electron Optics.175 pp 105–114. Siva Kumar Pathuri et al., International Journal of Emerging Trends in Engineering Research, 8(9), September 2020, 6380 – 6388 6388

[11] Sreedevi E, Premalatha, V, Sivakumar, S, Nayak, S.R 2019 A comparative study on new classification algorithm using NASA MDP datasets for software defect detection. Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2019. pp 312-317.

[12] V.PremaLatha, E Sreedevi, Sivakumar, S 2019 Contemplate on internet of things transforming as medical devices - The internet of medical things. (IOMT)Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2019. pp 276-281.

[13] Shaik Razia, P. Swathi Pryathyusha, N.Vamsi Krishna 2018 A Comparative study of machine learning algorithms on thyroid disease prediction. International Journal of Engineering and Technology (UAE). 7( 2.8). pp 315-319.

[14] Shaik Razia, M.R. Narasinga Rao 2017 Development and Analysis of Support Vector Machine Techniques for Early Prediction of Breast Cancer and Thyroid. JARDCS (Journal of Advanced Research in Dynamical and Control Systems). 9(6). pp 869-878.

[15] Pathuri SK, Anbazhagan N, Joshi GP, You J. Feature-Based Sentimental Analysis on Public Attention towards COVID-19 Using CUDA-SADBM Classification Model. Sensors. 2022; 22(1):80. <https://doi.org/10.3390/s22010080>.

[16] Videla, Lakshmi Sarvani, et al. "Deformable facial fitting using active appearance model for emotion recognition." Smart Intelligent Computing and Applications. Springer, Singapore, 2019. 135-144.

[17] Videla, Lakshmi Sarvani, et al. "Modified Feature Extraction Using Viola Jones Algorithm". Journal of Advanced Research in Dynamical and Control Systems.Volume 10, Issue 3 Special Issue, 2018, Pages 528-538.

[18] Videla, Lakshmi Sarvani and M. Ashok Kumar P. "Fatigue Monitoring for Drivers in Advanced Driver-Assistance System." IGI Global, 2020,

pp. 170-187.

[19] Atmakur, V.K., Siva Kumar, P.”A prototype analysis of machine learning methodologies for sentiment analysis of social networks”, International Journal of Engineering and Technology(UAE), 2018,pp 963-967.

[20] Prabha Selvaraj ,Vijay Kumar Burugari , D. Sumathi ,Rudra Kalyan Nayak ,Ramamani Tripathy,”Ontology based Recommendation System for Domain Specific Seekers”,Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) 2019,pp341.

[21] T. Shiva, T. Kavya, N. Abhinash Reddy, Shahana Bano.”Calculating The Impact Of Event Using Emotion Detection”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-7 May, 2019.

[22] Prasanth Kumar1\* Gera Pradeepini1. Pille Kamakshi,” Feature Selection Effects on Gradient Descent Logistic Regression for Medical. Data”, October 2019,International Journal of Intelligent Engineering and Systems 12(5) pp:278-286.

[23] Chintala, R. R., Narasinga Rao, M. R., \& Venkateswarlu, S. (2018). A proposal for observing conceived ladies having high risk of premature delivery using WHSN. International Journal of Engineering and Technology(UAE), 7(2), 53-56. doi:10.14419/ijet.v7i2.32.13524.

[24] Sajana, T., \& Narasingarao, M. R. (2018). An ensemble framework for classification of malaria disease. ARPN Journal of Engineering and Applied Sciences, 13(9), 3299-3307.

[25] Chintala, R. R., Narasinga Rao, M. R., \& Venkateswarlu, S. (2018). Review on the security issues in human sensor networks for healthcare applications. International Journal of Engineering and Technology(UAE), 7(2.32 Special Issue 32), 269-274.

[26] Deshpande, L., \& Rao, M. N. (2018). Concept drift identification using classifier ensemble approach. International Journal of Electrical and Computer Engineering, 8(1), 19-25. doi:10.11591/ijece.v8i1.pp19-25.

[27] Arba Asha Altaye, Dr. J. Sebastian Nixon,A Comparative Study on Big Data Applications in Higher Education.Volume 7, No. 12 December 2019,International Journal of Emerging Trends in Engineering Research.

[28] Gunawan Wang, Natanael Alamas, and Marcelina Anggraeni, The Use of Internet of Things and Big Data to Improve Customer Data in Insurance Company. Volume 7, No. 12 December 2019 International Journal of Emerging Trends in Engineering Research.

[29] Mane, S. U., & NarsingaRao, M. R. (2019). Large-scale compute-intensive constrained optimization problems: GPGPU-based approach.

doi:10.1007/978-981-13-0589-4\_54.

[30] Swetha, K., & Narasinga Rao, M. R. (2016). Dynamic searchable encryption over distributed cloud storage. Asian Journal of Information Technology, 15(23), 4763-4769. doi:10.3923/ajit.2016.4763.4769.