25(2) 2024

ISSN 2300-7036

COMPUTER SCIENCE

QUARTERLY OF AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

AGH

AGH UNIVERSITY PRESS

KRAKOW 2024

25(2) 2024 ISSN 2300-7036

COMPUTER SCIENCE

AGH UNIVERSITY PRESS

KRAKOW 2024

COMPUTER SCIENCE is published by AGH University Press, Krakow,
Poland.

The papers presented in COMPUTER SCIENCE have been accepted by the reviewers selected by the editors of the journal.

Head of Publishing of AGH University Press
Jan Sas

Technical Editor
Magdalena Grzech

Ghostwriting prevention
Łukasz Faber

Statistical Correction
Anna Barańska

Linguistic Correction
Bret Spainhour

Cover Design
Anna Sadowska

Typesetting and Desktop Publishing
Marek Karkula

ISSN 2300-7036
DOI: https://doi.org/10.7494/csci

Wydawnictwa AGH (AGH University Press)
al. A. Mickiewicza 30, 30-059 Kraków, Poland
tel. +48 12 617 32 28, +48 12 636 40 38
e-mail: redakcja@wydawnictwoagh.pl
https://www.wydawnictwa.agh.edu.pl

Shikha Gupta
Animesh Kumar

SENTIMENT-AWARE ENHANCEMENTS OF PAGERANK-BASED CITATION METRIC, IMPACT FACTOR, AND H-INDEX FOR RANKING THE AUTHORS OF SCHOLARLY ARTICLES

Abstract *Heretofore, the only way to evaluate an author has been frequency-based citation metrics that assume citations to be of a neutral sentiment. However, considering the sentiment behind citations aids in a better understanding of the viewpoints of fellow researchers for the scholarly output of an author. We present sentiment-enhanced alternatives to three conventional metrics namely Impact Factor, h-index, and PageRank-based index. The proposal studies the impact of the proposed metrics on the ranking of authors. We experimented with two datasets, collectively comprising almost 20,000 citation sentences. The evaluation of the proposed metrics revealed a significant impact of sentiments on author ranking, evidenced by a weak Kendall coefficient for the Author Impact Factor and h-index. However, the PageRank-based metric showed a moderate to strong correlation, due to its prestige-based attributes. Furthermore, a remarkable Rank-biased deviation exceeding 28% was seen in all cases, indicating a stronger rank deviation in top-ordered ranks.*

Keywords Kendall coefficient, rank biased distance, bibliometrics, scientometrics

Citation Computer Science 25(2) 2024: 173–210

1. Introduction

A research project begins with a concept, often culminating in a publication, typically in a conference or journal. The published article may mention other articles, mentions being called citations. While mentioning article is called the citing article, the mentioned article is called the cited article.

The study of the impact of authors and scholarly articles based on citation frequency is a popular area of research. There are two primary categorizations of citation-based approaches in the realm of evaluation of authors and articles. The first involves assessing the impact of an author/article as a function of citation counts. For instance, h-index is a well-known citation-frequency-based metric. Impact Factor is a metric that centers on total article count and total citation count values for an author/article. The second considers the prestige of the author/article. For instance, a citation metric based on the PageRank algorithm. It is assumed that each scholarly paper makes the same amount of scientific contribution and that each citation holds the same importance [34].

While the use of frequency-based indicators remains a common practice, it has been widely discussed that relying solely on the number of citations fails to truly measure the impact of the cited articles on the research domain [12,19]. For instance, an article cited just to discuss its shortcomings and improvements [30] may not be considered at par with one cited for positive contribution.

In recent years, a tangential focus while evaluating authors/articles has been to consider the content of a citation with the intent to understand the meaning behind citations, leveraging the inherent nature of academic writing [14]. Indeed, the importance of a citation changes based on its impact on the citing work – enhancing the cited work or providing background knowledge [32].

1.1. Motivation for the proposal

In the literature, the researchers have shown agreement against an over-reliance on pure quantitative metrics for measuring scientific impact [25]. In a recent study, Xu *et al.* [36] investigated the influence of qualitative aspects of a citation such as criticism or praise on research quality assessment by conducting a content analysis on top-rated papers from the Association for Computational Linguistics (ACL) conferences. The study concluded that the metric of citation polarity can help in better evaluation of the quality of research output and offer ideas for unbiased assessment of scholarly articles.

When examining citation motivation, negative citations are not solely indicative of criticism. Indeed, they frequently highlight the limitations and deficiencies within the referenced work. These shortcomings often serve as potential areas for improvement, suggesting that building upon the cited work may pave the way for enhancements.

However, it's crucial to acknowledge that while negative citations often signify limitations, their interpretation can be subjective and may not uniformly guarantee the inherent value or relevance of the cited material for the citing paper [36]. Considering citation polarity is, therefore, an important area of research. Taking into account the qualitative aspect of citations, specifically the sentiment's tone and polarity conveyed by the citing article, can facilitate a more precise assessment of an article's influence [36].

Even when there are several proposals for classifying citations according to their polarity and for computing the sentiment score of citations [1, 4, 5, 12, 14, 15, 17, 19, 20, 23, 25, 27, 29, 33, 38], only a few proposals [12, 16, 19, 21] address the problem of sentiment-aware citation metrics.

In addition, the literature is lacking in presenting sentiment-enhanced citation metrics for evaluation of authors of the scholarly articles and the current work offers a novel proposal.

1.2. Research objectives

In the present study, we consider the characterization of citations on the basis of the expressed sentiment as positive, negative, or neutral. Usually, the terms "positive" and "negative" for citations involve discerning the polarity of the citing article's opinion towards the cited article for a specific citation instance (praising or criticizing a specific aspect) [26].

The present work contributes to the literature by addressing the following research objectives:

- Objective 1: Propose a method to compute the 'Aggregated Sentiment Score' for an author of a scholarly article.
- Objective 2: Propose sentiment-aware enhancements based on the authors' aggregated sentiment score for the well-known citation metrics (h-index, Impact Factor, PageRank-based metric) to evaluate author impact.
- Objective 3: Study the impact on the ranking of authors obtained by the proposed sentiment-aware metrics vs. frequency-based citation metrics.

1.3. Our contributions

We aim to measure the impact of authors' contributions while relying on sentiment scores of the citation sentences. The set of citation sentences from citing articles is taken from the two datasets with pre-extracted citation contexts: Citation Sentiment Corpus [5] (Corpus 1) and Citation Intent Classification Dataset [9] (Corpus 2). The first dataset contains 8736 citations, and the second dataset contains 11020 citations, each along with their ground truth for sentiment.

After computing the ranks for authors and their articles using the proposed sentiment-based metrics, we conducted a comparison with their quantitative counterparts. Based on statistical analysis, our evaluation shows mostly weak but positive values for Kendall's Tau Coefficient when comparing the proposed metrics to traditional indices. It is worth noting that when measured using Rank-Biased Deviation (RBD), these deviations consistently exceeded 28%, highlighting the influence of the sentiment associated with citations. The statistical results are noted in the significant shifts in the author rankings for articles that have a strong sentiment, whether positive or negative. Remarkably, PageRank demonstrates moderate to high Kendall's Tau coefficient and a consistently low RBD among all indices, indicating that the inclusion of sentiment has a stronger impact when the metrics are based solely on popularity (frequency of citations).

In summary, the present proposal makes the following contributions:

- Considers sentiment polarity of the citations earned by the authors of scholarly articles in evaluating the author impact.
- Determines the sentiment score for each citation instance in a comprehensive collection of research articles. The experiments consider two datasets with almost 20,000 citation sentences.
- Develops a method for computing the aggregate sentiment score for the authors.
- Computes the overall sentiment score for each article and author in the collection.
- Proposes sentiment-aware alternatives to the well-known conventional citation metrics – h-index, Impact Factor, PageRank-based metric.
- Studies the impact of the proposed metrics on the ranking of the articles and the authors. The results are analyzed employing the following statistical measures: Kendall's Tau Rank Correlation and Rank Biased Distance (RBD).

1.4. Organization of the article

In the rest of the article, we review the related work in Section 2. The datasets are described in Section 3. Section 4 presents the steps in the proposed methodology. Section 5 details the experiments and their results. In response to the first research objective, Section 5.1 details our approach for identifying the sentiment in citations for the authors and the articles. The proposed sentiment-aware ranking metrics (research objective 2) are described in Section 5.2. To address the third research objective, the influence of the proposed citation metrics is evaluated in Section 5.3. A discussion of the results is presented in Section 6. Section 7 concludes the article and suggests a direction for future work.

2. Related work

While there's extensive research in the fields of Scientometrics, Sentiment analysis, and Network theory independently, only a limited number of studies exist at the intersection of the three domains, precisely where our work is positioned.

2.1. Sentiment analysis of citations

At the intersection of Sentiment analysis and Scientometrics is the sentiment analysis of citations, usually considered a two-step process, where the initial step identifies the explicit/implicit citation contexts and extracts their locations in the citing article. The subsequent step of assigning sentiment to citation instances often employs standard classifiers. In recent years, there's been a notable rise in automated sentiment classification of citations [1,4,5,12,14,15,17,19,20,23,25,27,29,33,38].

2.1.1. Identifying citation contexts

In scholarly articles, it is typically observed that either a citation is explicitly indicated in a complete sentence (termed as a citation sentence) ending with a citation mark or may be implicit with the reference details in the cited work extending beyond the explicit citation sentence, encompassing nearby sentences without citation marks [39].

Several tools are available to extract citations from research articles based on the article's structure and citation style. In recent studies by Nazir et al. [26] and Aljuaid et al. [3], citation sentences were extracted after converting PDFs to text using XPDF and then parsing these text files with ParCit. ParCit, an openly accessible tool, recognizes various structural elements in research articles, such as titles, authors, and abstracts, aiding in the extraction of citation sentences.

2.1.2. Sentiment-based classification of citations

In the popular methodologies for sentiment-based classification of citations, various machine learning and lexicon-based approaches have been employed. Usually, before employing machine learning classifiers for sentiment analysis, text preprocessing methods are applied. The following examples showcase varied preprocessing strategies in sentiment analysis research. One of the studies [12] utilized positive and negative polarity words, along with part-of-speech and dependency tags for feature extraction. In contrast, another study [27] employed lemmatization, n-grams, stop words removal, and term-document frequency. Meanwhile, a different study [31] implemented term frequency-inverse document frequency (TF-IDF) and Word2Vec techniques for data preparation and representation.

The landscape of citation sentiment analysis has undergone significant evolution parallel to advancements in sentiment analysis and machine learning algorithms. Popular methodologies include Decision Trees, SVM, Naive Bayes, Decision Trees, Random Forest, and KNN [27]. Some proposals [38] incorporate sophisticated models such as CNN and BiLSTM, reflecting a notable paradigm shift in methodology and analysis techniques.

2.1.3. Challenges in identifying citation's sentiment

Scientific texts present multiple hurdles for sentiment analysis, as sentiments are often concealed, written neutrally, and expressed through an objective style influenced by authors' biases [12]. The dual mode of writing, where criticism follows light appraisals, further complicates sentiment identification [22]. This complexity extends to other literary genres, also the variability in citation styles across publishers adds complexity to pinpointing citation locations as well as selecting the citation window size, especially when citations span multiple sentences [26].

A challenge encountered in sentiment-based classification of citations arises from significant bias within datasets, notably towards the neutral class. The bias towards the neutral class leads to class imbalance issues, causing erroneous predictions and potentially overfitting machine learning models [38]. Accurately detecting subtle sentiments, especially refined negative ones in citations, presents an ongoing challenge. This difficulty stems from their implicit nature and veiled criticisms, posing a significant hurdle for both algorithms and human perception [12, 19].

Another challenge arises from technical terms or specialized jargon in research areas, devoid of inherent sentiment, yet causing noise in sentiment score computation [37]. For example, the term 'Support' in 'Support Vector Machines' implies a positive sentiment, thereby complicating the sentiment analysis process.

Since the focus of the current work is on proposing sentiment-infused citation metrics for author ranking, we employ two corpora of pre-extracted citation instances. The popular lexical resource SentiWordNet [12, 19] is employed for deciding the sentiment scores of the citation sentences. Therefore, the proposed approach is not required to address the above challenges in its approach.

2.2. Assigning sentiment score to a citation instance

Apart from supervised machine learning algorithms, some lexicon-based resources have also evolved that provide sentiment scores for sentences followed by categorization using the scores. A citation with a "Positive" or a "Negative" polarity does not necessarily indicate "good" or "bad". While a citing article may criticize a specific aspect of a cited article, it may praise a different aspect of the same cited article. Therefore, sentiment scores are employed instead of simply classifying citations based on pure sentiment.

Many researchers have proposed the use of sentiment scores that are imbued with both sentiment and degree. Sentiment scores involve determining the sentiment of the citing paper towards the cited paper for a specific citation instance – appreciating a particular aspect or criticizing the same [12]. A study has employed SenticNet [37] that integrates concepts and semantics to offer polarity and semantic information for concepts. More popularly, SentiWordNet [12, 19] is employed as a lexical resource for identifying citation sentiment based on the generated sentiment score. SentiWordNet is a lexical resource that associates sets of cognitive synonyms (synsets) with sentiment scores (positive, negative, neutral).

2.3. Influence of citation sentiments on scientific articles

At the intersection of Scientometrics, Network theory, and Sentiment analysis lies research in citation content analysis offering a nuanced and qualitative perspective on a cited article [17]. In the past decade, the field of scientometrics has seen a rise in the thought that an article's impact or usefulness shouldn't solely rely on the number of times it's cited but also on the manner of its citation. This means that the opinions expressed by the citing authors can serve as a valuable gauge of an article's influence. However, most of the existing bibliometric measures to evaluate scholarly articles and their authors are majorly quantitative with a focus on the number of times an article is cited, based on a prevailing assumption that research articles are generally cited in a positive manner [12]. Abu-Jbara et al. (2013) [1] critiqued conventional bibliometric measures for lacking the ability to distinguish between positive and negative citations. However, few studies can be found that consider sentiment when assessing the impact of scholarly articles [12,16,19,21]. Moreover, we did not find any prior work that ranked and evaluated the authors based on the sentiment analysis of citations.

Kazi et al. [16] proposed to incorporate semantic similarity considering the same between citing and cited article along with the polarity of the text surrounding the citation sentence and self-citations. The authors employ SentiWordNet to generate sentiment scores and show that their results are parallel to those of the PageRank-based approaches and are an improvement over traditional citation counts.

Ma et al. [21] proposed a method to classify the citations by integrating the sentiment polarity with the data about the authors' reputation, encoded in the form of p-index. p-index is obtained by multiplying the h-index value by the number of positive citations raised by a value greater than 1 and number of negative citations raised by a fractional power. However, the authors do not give the rationale behind the formulation of the equation for computation of p-index. The authors concluded that integrating the reputation of the authors in the input data significantly improves the process of citation sentiment classification.

Kochhar and Ojha [19] proposed an equation-based sentiment-enhanced impact factor for articles, which incorporates multiple factors. The approach considers sentiment scores (computed using SentiWordNet), the impact factor of both the citing and cited authors, and the respective publishing journals. The authors concluded that it is important to consider the sentiment behind the citations instead of simply considering that each citation contributes equally to the evaluation of the article's impact.

Ghosh and Shah [12] highlighted the importance of considering sentiment of citation and presented ranking indexes to appraise the significance of research papers. Based on supervised ML (machine learning) classifier, the authors determine the polarity of a citation sentence. Thereafter, the classifier is used to assign sentiment scores to citation instances. Their sentiment-based metric to rank articles uses overall citation score obtained by summing up the citation scores from citing to cited papers. The authors also proposed that the PageRank-based approach to rank articles be modified by considering the associated sentiment while determining the edge weight. For any edge from citing to cited article, the score transferred is multiplied by the sentiment score of the citation instance.

We present a comparative analysis of articles focusing on sentiment-aware metrics in Table 1. This table offers insights into the reviewed literature, highlighting methodology, results, and limitations among various studies in the field of sentiment-aware citation metrics.

We compare our work with the study by Ghosh and Shah [12]. As compared to others, the work by [12] comes nearest to our proposal. We mark the following differences:

- The focus of the present work is on proposing sentiment-aware citation metrics for ranking authors. Ghosh and Shah [12] have focused on ranking articles.
- Ghosh and Shah [12] identify the sentiment in each citation sentence using supervised machine learning classifier. The present approach preprocesses the citation sentences to make the data suitable for input to SentiWordNet. The sentiment scores are computed using the widely used SentiWordNet. The proposed technique precludes any bias that may arise from technical terms and class imbalance issues.
- The cited article may be mentioned more than once in the text of the citing article. For every pair of citing and cited author/article, we associate a composite score representing the citation frequency and sentiment score values. Moreover, the context of the cited article/author is aggregated and an "aggregate sentiment score" is assigned to the authors and articles. Ghosh and Shah [12] use the classifier to assign sentiment scores to citation instances.
- We propose sentiment-infused alternatives to the three well-known conventional frequency-based citation metrics for ranking authors – h-index, Impact Factor, PageRank-based metric. Ghosh and Shah [12] propose sentiment-based modifications for following metrics to rank articles – citation count and PageRank-based approach.
- The proposed approach incorporates in its PageRank-based approach, the aggregate sentiment score as the edge weight. On the other hand, Ghosh and Shah [12] proposed that the PageRank-based approach to rank articles be modified by associating the sentiment to the edge weight. The authors [12] mention that for any edge from citing to cited article, the score transferred is multiplied by the sentiment score of the citation instance.
- The proposed approach clearly defines the damping factor value appropriate to the domain for its PageRank-based approach. The value of damping factor is significant in the working of PageRank algorithm as it can induce notable deviations even with minor adjustments [7]. Conversely, Ghosh and Shah [12] do not discuss their selection of a damping factor value.

Table 1

Related literature for the primary objective of proposing sentiment-enhanced citation metrics for ranking of authors

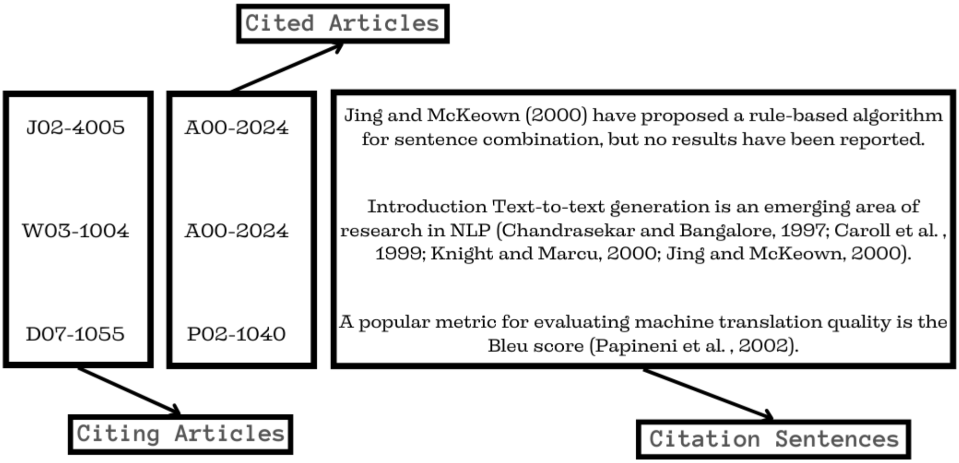
Article	Experiments	Results	Limitations
[12]	1. Extracted features from the dataset and employed a meta-classifier (DAGGING) to facilitate automatic annotation of an unlabeled corpus. 2. Above corpus was used for computing the ranks using the proposed indexes. 3. Evaluated the performance of the generated indexes by comparing each ranked list with others. The evaluation criterion involved identifying differences between the order of ranked lists	1. The overall accuracy of the classifier came out to be 80.61%. 2. Inclusion of sentiment and link structure led to more than 25% of difference between two ranking indexes	1. The methodology explains the training and application of the sentiment classifier but lacks details on the specific computation process for generating sentiment scores. 2. The study defines self-citations as instances where the Source paper and Target paper are the same. However, self-citation often extends to the author's previous works rather than being limited to the same paper. 3. The study does not explicitly state the chosen damping factor for the PageRank indices, which is a crucial parameter in PageRank algorithms. 4. Additionally, the research scope is limited to analyzing articles, neglecting exploration into authors or journals
[19]	1. Extracted citations from the AAN repository and preprocessed citation contexts for analysis. 2. Calculated sentiment scores for each citation instance using SentiWordNet. 3. Computed metric values based on a defined equation, incorporating publication and authors' impact factors	The rankings were determined by utilizing calculated impact factors for ten articles, with a primary emphasis on a broader set of 91 papers. This collection comprised the initial ten papers, along with the additional papers that cited them	1. Lack of detailed discussion regarding the handling of multiple citations between a pair of papers and self-citations. 2. The final results (ranks) were not systematically assessed for trends or deviations, limiting a comprehensive analysis of the experimental outcomes. A small dataset is considered. 3. The research scope is limited to analyzing articles, neglecting exploration into authors or journals

Table 1 cont.

Article	Experiments	Results	Limitations
[16]	1. Computed sentiment scores using SentiWordNet, along with weighted self-citations and semantic similarity values. 2. Utilized the computed values to assign scores to citation edges in the network and applied a graph traversal method to calculate the citation index. 3. Evaluated the correlation between PageRank algorithm ranks and those obtained through the graph traversal approach	1. The Spearman coefficient between the Graph Traversal Method and Weighted PageRank yielded a highly significant value of 0.995	1. When assigning weights to various factors to compute edge weights, there is a comparatively lesser level of consideration. 2. Although Spearman is a commonly employed correlation metric, its robustness diminishes when computing correlation in scenarios with a substantial number of ties. 3. The research area diverges from ours as it focuses on an alternative method for computing PageRanks in citation networks
[21]	1. Derived an index (p-index) by incorporating sentiment into the h-index through a dedicated equation. 2. Utilized the newly created p-index and additional author reputation information as extensions to the fundamental features for classification purposes. 3. Trained and tested an SVM model to assess the impact on accuracy resulting from the integration of the p-index and supplementary author reputation details	1. It was noted that the combined inclusion of Affiliation ID and Author ID yielded enhanced performance	1. The method for counting the number of positive and negative citations to compute the p-index lacks an explicit description. 2. The formulation of the equation for the p-index lacks thorough discussion and clarification. 3. Their research deviates from ours as they emphasize the utility of the derived index for enhancing the classification of citation sentiment

3. Data description

We conducted sentiment analysis on scientific citations using two corpora [5, 9] comprised of citation sentences. As can be seen in Figure 1, the sentence containing the citation (for the "cited article") is called the "Citation Sentence". The article containing the citation sentence is referred to as the "citing article".



Jing and McKeown (2000) have proposed a rule-based algorithm for sentence combination, but no results have been reported.

Introduction Text-to-text generation is an emerging area of research in NLP (Chandrasekar and Bangalore, 1997; Caroll et al. , 1999; Knight and Marcu, 2000; Jing and McKeown, 2000).

A popular metric for evaluating machine translation quality is the Bleu score (Papineni et al. , 2002).

Figure 1. Examples of Citation Instances from Corpus 1

In tandem with our focus, we searched for datasets that provided pre-extracted citation sentences. The Citation Sentiment Corpus by Athar (2011) [5] is a widely employed dataset in the field of citation sentiment analysis and includes citation contexts extracted from both citing and cited papers within the AAN repository. This dataset (Corpus 1) is from the ACL Anthology, a repository renowned for housing research articles in computational linguistics. An instance in the dataset contains a single sentence. The dataset comprises 8736 citation instances. Figure 1 shows a sample from the dataset. The dataset contains the following information: Citation sentence, citing article ID, the cited article ID, and the sentiment of a citation.

Comparison revealed that other available datasets have a lower number of citation instances. For example, Malakwani Ram's [23] dataset yielded approximately 5000 instances, Yousif's [38] approximately 2000 instances, and other existing datasets, such as Valenzuela's [32], contained around 500 instances.

The second dataset (Corpus 2) comprised 11,020 instances and was sourced from the Scicite intent corpus [9]. The corpus was created from a subset of the Semantic Scholar corpus [1]. Since it was initially designed for intent classification, we extracted the necessary fields, including citing article ID, cited article ID, and citation instances.

[1] https://semanticscholar.org/

The rationale for selecting this dataset was driven by its inherent benefit wherein utilizing article IDs enabled the efficient extraction of the author and other article details through the Semantic Scholar API. In addition, since the API supplies author IDs, author name disambiguation is not needed.

In addition, SciCite Intent Corpus (Corpus 2) demonstrated the impact of sentiment-aware metrics, particularly with a small number of citations. As shown in further sections, the sentiment-infused metrics serve as effective differentiators, functioning as tiebreakers, to efficiently distinguish between articles with minimal differences in citation counts. For instance, consider the citation sentence: "Our data also showed that lesions in the head and neck region responded better to PDL therapy than in other regions; similar results were reported by several studies [16,24,28]." This sentence attributes a citation to the author identified by the author ID "10616537," who has received a singular citation, thereby placing the author at the shared rank of 11588 with several others, including author ID "3881197." However, an examination incorporating sentiment analysis yielded a score of "0.75" and "−1.5" and substantially different corresponding ranks of 2681 and 23267 respectively for the authors. A detailed comparative analysis is expounded upon in Section 6.

4. Proposed method

The focus of the present work is on proposing sentiment-aware citation metrics for ranking authors. We use datasets with pre-extracted citation contexts. A flowchart to outline the steps involved in the proposed method is presented in Figure 2.

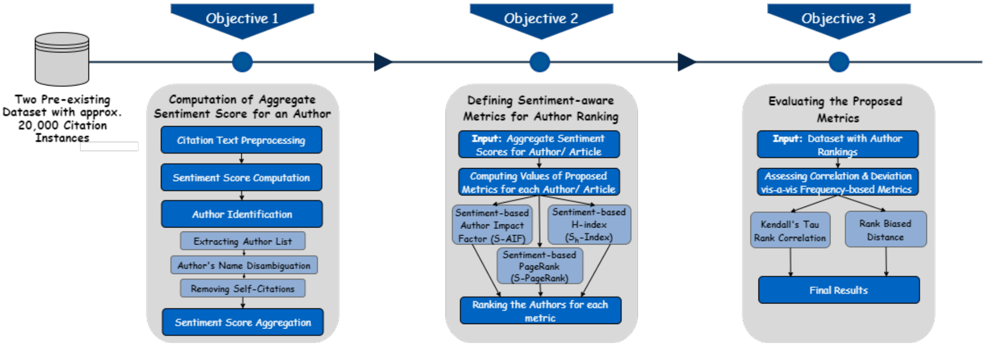

Objective 1
Objective 2
Objective 3
Two Pre-existing Dataset with approx. 20,000 Citation Instances
Computation of Aggregate Sentiment Score for an Author
Citation Text Preprocessing
Sentiment Score Computation
Author Identification
Extracting Author List
Author's Name Disambiguation
Removing Self-Citations
Sentiment Score Aggregation
Defining Sentiment-aware Metrics for Author Ranking
Input: Aggregate Sentiment Scores for Author/ Article
Computing Values of Proposed Metrics for each Author/ Article
Sentiment-based Author Impact Factor (S-AIF)
Sentiment-based H-index (S_h-Index)
Sentiment-based PageRank (S-PageRank)
Ranking the Authors for each metric
Evaluating the Proposed Metrics
Input: Dataset with Author Rankings
Assessing Correlation & Deviation vis-a-vis Frequency-based Metrics
Kendall's Tau Rank Correlation
Rank Biased Distance
Final Results

Figure 2. Flowchart detailing the steps involved in the proposed method

Following are the broad steps that the proposed method follows:

- Step 1 corresponds to the first research objective (Section 5.1). We pre-process the citation sentences to make the data suitable for input to SentiWordNet. The sentiment scores are computed using the widely used SentiWordNet which prevents any bias that may arise from technical terms and class imbalance issues. We assign a composite score, which we call the "Aggregate Sentiment Score", to the authors and articles. This entailed identifying the unique authors in the dataset and removing self-citations before aggregating the scores for each author.

- Step 2 corresponds to the second research objective (Section 5.2). We propose sentiment-aware alternatives to the three well-known conventional frequency-based citation metrics – h-index, Impact Factor, PageRank-based metric. Each of the three proposed metrics replace use of simple citation frequency with the use of the composite sentiment score. To the best of our knowledge, this is first such proposal for ranking authors.
- In the end, corresponding to the third research objective (Section 5.3), we study the impact on author rankings of the proposed sentiment-aware metrics compared to the frequency-based citation metrics.

5. Experiments and results

In the current section, we present the experimentation performed and the results derived thereof. We have divided the same into three subsections, corresponding to our three objectives:

1. Computation of the aggregate sentiment score for an author.
2. Proposal for sentiment-aware metrics for evaluating the authors of scholarly articles.
3. Evaluation of the proposed metrics for ranking authors.

5.1. Objective 1: computation of aggregate sentiment score for an author

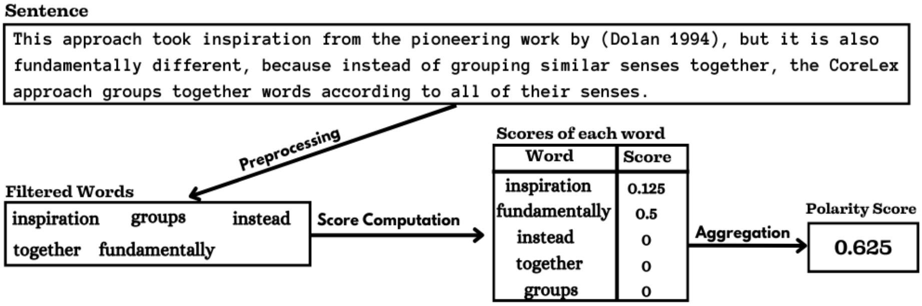
The process starts by preprocessing the citation sentences (Section 5.1.1) following which we utilized SentiWordNet [6] to compute the sentiment scores for each citation sentence.

SentiWordNet is a specialized version of WordNet [10], a lexical database for the English language. WordNet provides definitions based on the part of speech and synsets – groups of synonyms conveying the same concept. Notably, SentiWordNet goes beyond WordNet, offering numerical scores for each synset.

SentiWordNet automatically annotates the synsets of WordNet and marks a synset (say, syn) with a three-dimensional value (syn_{+}, syn_{-}, $syn_{=}$) to indicate its degree of "positivity", "negativity", and "neutrality", The scores for each synset are distributed in a manner that ensures the cumulative sum equals 1, as depicted in the following Equation (1):

$$syn_{+} + syn_{-} + syn_{=} = 1,\ 0 \leq syn_{+}, syn_{-}, syn_{=} \leq 1 \tag{1}$$

The proposed method provides a quantitative measure of sentiment for a citation sentence, based on the total sentiment score of its constituent words. For computing the score of a citation sentence, the sentiment scores of each synset in a citation sentence are added. Figure 3 shows an example for calculating the sentiment score of a citation sentence.



This approach took inspiration from the pioneering work by (Dolan 1994), but it is also fundamentally different, because instead of grouping similar senses together, the CoreLex approach groups together words according to all of their senses.

inspiration groups instead together fundamentally

Word	Score
inspiration	0.125
fundamentally	0.5
instead	0
together	0
groups	0

Figure 3. Example showing the computation of the sentiment score of a citation sentence

A cited article may be referenced in the citing article once or multiple times, each instance being termed as a citation sentence. Usually, each instance of a citation in a citing article is evaluated separately for its polarity. A function of the individual citation sentiments can be computed to ascertain the overall polarity [12]. We evaluated sentiment scores for individual citation sentences and added them to establish the overall sentiment score related to each cited article.

After computing the scores for each cited article, the subsequent step was to derive the scores for the authors (Section 5.1.2). This entailed identifying the unique authors in the dataset and removing self-citations before aggregating the scores for each author.

5.1.1. Preprocessing and sentiment-based scoring of citation text

Preprocessing is a crucial component in enhancing the quality of the text data before sentiment computation. The following steps of preprocessing set the stage for a more effective and accurate sentiment score computation, aligning with the standards of scientific research:

- **Tokenization** was performed to breakdown the sentences into list of words.
- **Part-of-speech (POS) Tagging** was used to assign grammatical attributes to each token. We specifically retained lemmas categorized as adverbs, adjectives, and nouns, known for carrying sentiment-related information [19].
- **Lemmatization** simplified subsequent analysis by reducing words to their base form. Lemmatization was then done based on the POS Tags using WordNet to get the respective lemma for each word.
- We applied a **counter-based curation strategy** to remove frequently occurring lemmas that might represent domain-specific or neutral terms, minimizing potential interference with sentiment analysis [24].

Score Computation: We created a Python program utilizing key libraries including NLTK, Pandas, and Requests. To gather essential author information for Corpus 1, we utilized a Python script with the requests library to fetch data from the ACL Anthology, a repository renowned for housing research articles in computational linguistics. The SentiWordNet functionality of the NLTK library was employed to compute sentiment scores for individual citation sentences. It is noted that a citation sentence comprises several lemmas and multiple citation sentences may exist corresponding to a cited article in the text of a single citing article.

Consider a cited article, say 'A' mentioned m times in a citing article, say 'B'. Assume that the j^{th} citation sentence identified n_j lemmas where $j \in [1,\ m]$. For the i^{th} lemma in the j^{th} citation sentence (say, $syn^i(j),\ i \in [1,\ n_j]$) with $syn^i_+(j)$, $syn^i_-(j)$, and $syn^i_=(j)$ as positivity, negativity, and neutrality scores, the total sentiment score for the cited article 'A' w.r.t the citing article 'B' is computed as presented in Equation (2):

$$\text{Total Sentiment Score} = \sum_{j=1}^{m}\sum_{i=1}^{n}(syn^i_+(j) - syn^i_-(j)) \tag{2}$$

That is, the positive and negative sentiment values are given a weight of "1" and "−1" respectively. The neutrality score is given a weight of "0". This is done to ensure that even a small amount of sentiment score is registered in the "Total Sentiment Score". Since the scientific text mostly contains neutral words, giving a non-zero weight to the neutrality score can overshadow the positive or negative opinion of the citing author.

Additionally, using the above weights, a neutral synset will obtain a "Total Sentiment Score" of 0. Therefore, for an article cited without sentiment implication, sentiment-enhanced and traditional frequency-based metrics will lead to a similar evaluation.

5.1.2. Author identification and score aggregation

Our primary emphasis revolved around examining author ranking based on sentiment. To achieve this, we utilized the distinct article IDs within the datasets, subsequently extracting the authors associated with each article.

Extracting Author List: Our objective was to automate the compilation of a list of authors for each distinct article in our dataset. To accomplish this for Corpus 1, we dynamically constructed the URL for each article by adding its unique identifier to the ACL Anthology's base URL [2] and retrieved the webpage. After successfully fetching the webpage, we employed regular expressions to parse the HTML content and extract the necessary author information. In some cases, the URL was unavailable. In such instances, we conducted manual searches using the corresponding cited/citing articles, and examined the citation sentences to extract the details of these articles. Table 2 shows the extracted author names corresponding to a subset of the articles in Corpus 1.

For Corpus 2, we automated the communication with the Semantic Scholar API [18] to retrieve Author IDs for each article. In some cases where the author

[2] https://aclanthology.org/

IDs could not be obtained via the Semantic Scholar API, we manually searched using the citation sentences. Even after the manual search, the Author IDs could not be retrieved for 65 articles, leading us to exclude them from subsequent processes.

Table 2

The extracted author names for a subset of the articles in Corpus 1

Article	Author1	Author2	Author3	Author4
A00-1004	Chen, Jiang	Nie, Jian-Yun	–	–
A00-1005	Bagga, Amit	Strzalkowski, Tomek	Wise, G. Bowden	–
A00-1007	Jonsson, Arne	Dahlback, Nils	–	–
A00-1011	Aone, Chinatsu	Ramos-Santacruz, Mila	–	–
A00-1012	Stevenson, Mark	Gaizauskas, Robert	–	–
A00-1014	Chu-Carroll, Jennifer	–	–	–
A00-1019	Langlais, Philippe	Foster, George	Lapalme, Guy	–
A00-1020	Harabagiu, Sanda	Maiorano, Steven J.	–	–
A00-1025	Cardie, Claire	Ng, Vincent	Pierce, David	Buckley, Chris
A00-1026	Rindflesch, Thomas C.	Rajan, Jayant V.	Hunter, Lawrence	–
A00-1031	Brants, Thorsten	–	–	–
A00-1039	Yangarber, Roman	Grishman, Ralph	Tapanainen, Pasi	–
A00-1042	Wacholder, Nina	Klavans, Judith L.	Evans, David K.	–
A00-1043	Jing, Hongyan	–	–	–

Author Disambiguation: After the initial extraction of the authors' names, we considered the following potential scenarios as motivation to disambiguate the author names:

1. Two or more (slightly different) author names may refer to the same individual.
2. The same name may be attributed to two different authors.

To address the first condition in Corpus 1, the authors were organized based on their last names and an investigation of potential similarities in the spelling and abbreviations of adjacent names was carried out. We compiled a list of names that could be ambiguous and searched the ACL Website, specifically in the authors' profiles under the "Also published as" section of the website. A few examples of authors along with their alternate names, identified during this disambiguation process, are as below:

- Benedi, Jose-Miguel & Benedí, J.M.;
- Biermann, Alan & Biermann, Alan W.;
- Cai, Junfu & Cai, Jun Fu;
- Church, Ken & Church, Kenneth & Church, Kenneth W.;
- Penstein-Rosé, Carolyn & Rosé, Carolyn & Rosé, Carolyn P. & Rosé, Carolyn Penstein.

Following the above step, we refined the list of authors to eliminate any ambiguities corresponding to the first condition. For the second condition, we utilized the updated author list and performed searches for each author on the ACL website, focusing on the presence of the "Other people with similar names" section of the website. Our investigation revealed that there were no instances of the second condition in Corpus 1.

In the case of Corpus 2, since the dataset allowed us to extract the Author IDs rather than their names, it obviated the need for the author disambiguation process.

Self Citations: Self-citations occur when an article cites another article with which it shares at least one author. In our study, we carefully examined the dataset to find such instances of self-citation. To do so, a Python script was developed. The script identified matching authorship in a given pair of citing and cited papers and extracted the corresponding Article IDs. For instance, 23, and 18 self-citations were found for the authors "Collins, Michael" and "Koehn, Philipp" (Corpus 1) respectively. Subsequently, a Python script was utilized to remove all instances of citations associated with the found pairing of citing and cited articles from the dataset to ensure that self-citations did not influence our further research.

Computation of Aggregate Author Score: We proceed to consolidate the "Total Citing articles" and "Total Sentiment Score" for each author within our study. The aggregation process of the sentiment score involved the summation of sentiment scores linked to the citations of the author. Table 3 displays 12 citation sentences and sentiment scores directed toward two articles authored by Kim, Soo-Min. These citations originate from seven distinct citing articles. Table 3 shows the process of computing the aggregate sentiment score for the example author to clarify the process for the same.

Table 3

List of citation sentences, their sentiment scores, and the sentiment Score per cited article for the Author "Kim, Soo-Min". Two author's two articles are cited seven times. The "Aggregated Sentiment Score" for the author is −7

Cited Article	Citing Article	Citation Context	Sentiment Score/ Sentence	Sentiment Score/ Cited Article
D07-1113	C08-1052	As well as the sentiment expressions leading to evaluations, there are many semantic aspects to be extracted from documents which contain writers opinions, such as subjectivity (Wiebe and Mihalcea, 2006), comparative sentences (Jindal and Liu, 2006), or predictive expressions (Kim and Hovy, 2007)	−0.5	−3
	C08-1060	Specifically, Kim and Hovy (2007) identify which political candidate is predicted to win by an opinion posted on a message board and aggregate opinions to correctly predict an election result	−0.875	
		Opinion forecasting differs from that of opinion analysis, such as extracting opinions, evaluating sentiment, and extracting predictions (Kim and Hovy, 2007)	−1.875	
		Kim and Hovy (2007) make a similar assumption	0	
	C08-1101	An application of the idea of alternative targets can be seen in Kim and Hovys (2007) work on election prediction	0.25	
	P09-1026	Kim and Hovy (2007) predict the results of an election by analyzing forums discussing the elections	0	

Table 3 cont.

Cited Article	Citing Article	Citation Context	Sentiment Score/ Sentence	Sentiment Score/ Cited Article
W06-0301	C08-1103	(2005), Kim and Hovy (2006)), source extraction (e.g. Bethard et al.)	0	−4
		A notable exception is the work of Kim and Hovy (2006)	0.375	
	D07-1114	In open-domain opinion extraction, some approaches use syntactic features obtained from parsed input sentences (Choi et al., 2006; Kim and Hovy, 2006), as is commonly done in semantic role labeling	−0.625	
		Kim and Hovy (2006) proposed a method for extracting opinion holders, topics and opinion words, in which they use semantic role labeling as an intermediate step to label opinion holders and topics	−1.875	
		Open-domain opinion extraction is another trend of research on opinion extraction, which aims to extract a wider range of opinions from such texts as newspaper articles (Yu and Hatzivassiloglou, 2003; Kim and Hovy, 2004; Wiebe et al., 2005; Choi et al., 2006)	−1.875	
	W07-2072	Kim and Hovy (2006) integrated verb information from FrameNet and incorporated it into semantic role labeling	0	

5.1.3. Results for objective 1

Tables 4 and 5 present the results for the articles and authors, respectively, based on the number of citing articles and the total sentiment score value. For the purpose of discussion, we have focused on the results for the top 20 articles and authors from Corpus 1.

Among the top 20 articles (Tab. 4), articles such as "J93-2004", "P02-1040", and "N03-1017" exhibit a rank deviation of a maximum of two positions between rankings computed based on sentiment score and citation frequency while attaining high ranks (top 5) in both. However, it is noteworthy that articles labeled "P03-1021" and "C94-2113" exhibit a substantial difference of 174 (negative sentiment) and 53 ranks (positive sentiment) respectively, between their citation-based and sentiment-based rankings.

Among the top 20 authors (Tab. 5), "Marcus, Mitch", "Collins, Michael", "Papineni, Kishore", "Roukos, S.", "Ward, Todd", "Zhu, Wei-Jing", "Berger, Adam L.", and "Wu, Dekai" show a rank deviation of a maximum of two positions between sentiment and citation ranking. Conversely, authors "Carletta, Jean", "Brown, Peter F.", "Mercer, Robert L.", and "Dolan, Bill" display notable discrepancies of 20 ranks or more with a maximum variation via positive sentiment of 94 ranks for "Dolan, Bill". This assessment highlights the significant influence of both positive and negative sentiments on deviations in rankings.

Table 4
Top 20 articles (Corpus 1) by number of citations and by sentiment score value

Article ID	# of Citing Articles	Sentiment Score	Rank by Citation	Rank by Sentiment
J93-2004	434	62.125	1	1
J93-2003	366	4.875	2	25
P02-1040	303	34.94	3	5
P03-1021	271	−7.584	4	178
N03-1017	218	41.25	5	3
J96-1002	212	26	6	6
P97-1003	177	35.625	7	4
W96-0213	166	10.5	8	17
P95-1026	151	10.625	9	16
J97-3002	147	19.125	10	8
J93-1003	138	18.25	11	10
J96-2004	121	60.75	12	2
J92-4003	117	2.125	13	57
W95-0107	109	4.875	14	25
W02-1001	107	7.875	15	20
J90-1003	99	11.5	16	15
P02-1053	94	22.375	17	7
J93-1007	71	3.125	18	44
P90-1034	67	13	19	14
W02-1011	66	18.25	20	10
Remaining articles in top 20 by Sentiment Score				
C94-2113	9	18.75	62	9
P04-1035	59	17.75	22	12
N03-1003	49	16	25	13
P06-1101	24	9.75	29	18
N06-1020	24	8.625	29	19
C98-2122	35	7.875	26	20

Table 5
Top 20 authors (Corpus 1) by citations

Author Name	# Citing Articles	Sentiment Score	Rank by	
			Citation	Sentiment
Della Pietra, Vincent J.	702	34.625	1	15
Della Pietra, Stephen	585	32.5	2	16
Marcus, Mitch	543	67	3	1
Och, Franz Josef	512	39.541	4	9
Brown, Peter F.	490	8.625	5	34
Mercer, Robert L.	490	8.625	5	34
Santorini, Beatrice	434	62.125	7	2

Table 5 cont.

Author Name	# Citing Articles	Sentiment Score	Rank by	
			Citation	Sentiment
Marcinkiewicz, Mary Ann	434	62.125	7	2
Collins, Michael	334	47.75	9	7
Papineni, Kishore	303	34.94	10	11
Roukos, S.	303	34.94	10	11
Ward, Todd	303	34.94	10	11
Zhu, Wei-Jing	303	34.94	10	11
Marcu, Daniel	283	52.375	14	6
Koehn, Philipp	241	45.5	15	8
Berger, Adam L.	212	26	16	17
Lee, Lillian	181	53.125	17	5
Ratnaparkhi, Adwait	166	10.5	18	32
Wu, Dekai	162	23.5	19	19
Yarowsky, David	151	10.625	20	31
Remaining authors in top 20 by Sentiment Score				
Carletta, Jean	121	60.75	24	4
Pang, Bo	132	37.125	22	10
Turney, Peter	103	23.75	27	18
Dolan, Bill	9	18.75	114	20

5.2. Objective 2: sentiment-aware metrics for evaluating authors

The influence of authors/articles within the scholarly community can initially be gauged by establishing a citation network, with authors/articles connecting through citations. In a citation network, an edge is directed from the citing to the cited author or article. Figure 4 shows a subset of the authors' citation network created for our analysis with nodes representing authors of scholarly articles.

The citation network analysis brings into focus the importance of articles, authors, and ideas while identifying their interconnectedness and helps pinpoint the most influential authors/articles. Moreover, the absence of citations can be a valuable hint, revealing unexplored domains or novel ideas for future research [12].

Each citation serves as a prime candidate for sentiment analysis. For the citation network, this entails assigning sentiment-based edge weights to the directed edges in the network [12]. For example, in Figure 4, the edge from author "Liu, Qun" to author "Brown, Peter F." displays the weight $3(-0.5)$, implying three citations and a negative sentiment score of 0.5.

As discussed in the earlier sections, after a pre-processing of citation texts in the corpora, author identification and sentiment score aggregation for articles and authors were performed. The proposed approach employed "SentiWordNet" [6] to derive sentiment scores for individual instances.

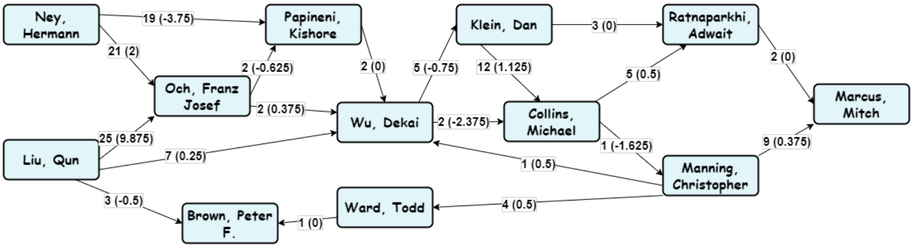


Figure 4. A subset of the author citation network (Corpus 1)

The proposed approach then establishes a pair of directed networks for the authors; one centered around citations and the other on sentiment. In the citation network, authors were represented as nodes, while the edge weights symbolized the cumulative count of citations exchanged between cited authors and citing authors. As for the sentiment-based network, it incorporated the overall sentiment score of citations to offer a fresh perspective on the ranking of the authors.

We developed three distinct alternatives for ranking based on sentiment: 1) Sentiment-based Author Impact Factor (S-AIF), 2) Sentiment-based h-index (S_h-index), and 3) Sentiment-based PageRank (S-PageRank) for author ranking. These alternatives are explained in the subsequent sub-sections.

5.2.1. Sentiment-based Author Impact Factor (S-AIF) and its results

The Author Impact Factor (AIF) quantifies an author's influence in academia by measuring the citations their work receives [11]. The author's impact can be computed as the total citations (NC) to an author's publications divided by the total number of publications (NP), the formula for calculating the Author Impact Factor is expressed in Equation (3). In its calculation, AIF considers the citation frequency and does not include the notion of citation quality.

$$AIF = \frac{\text{NC}}{\text{NP}} \tag{3}$$

In our study, we evaluate a different approach by utilizing the Total Citation Sentiment Score (S-NC), a measure of the quality of citations, departing from the traditional use of the number of citing articles. The proposed sentiment-based Author Impact Factor (S-AIF) is a score that takes into account the sentiment conveyed by citations, offering a more refined evaluation of an author's impact within academia. The formula for calculating the Sentiment-based Author Impact Factor is expressed in Equation (4). In its calculation, AIF considers the citation frequency and does not include the notion of citation quality.

$$S - AIF = \frac{\text{S-NC}}{\text{NP}} \tag{4}$$

Table 6 presents the results of the top 20 authors (Corpus 1) by AIF and by S-AIF.

Table 6
Top 20 authors (Corpus 1) by AIF

Author	AIF	S-AIF	Rank by	
			AIF	S-AIF
Santorini, Beatrice	434.00	62.13	1	1
Marcinkiewicz, Mary Ann	434.00	62.13	1	1
Zhu, Wei-Jing	303.00	34.94	3	3
Berger, Adam L.	212.00	26.00	4	4
Della Pietra, Vincent J.	175.50	8.66	5	10
Brown, Peter F.	163.33	2.88	6	27
Della Pietra, Stephen	146.25	8.13	7	11
Dunning, Ted	138.00	18.25	8	5
Mercer, Robert L.	122.50	2.16	9	38
deSouza, Peter V.	117.00	2.13	10	39
Ward, Todd	75.75	8.74	11	9
Hindle, Donald	67.00	13.00	12	6
Lai, Jenifer C.	62.00	1.88	13	45
Marcus, Mitch	60.33	7.44	14	13
Cutting, Doug	60.00	3.75	15	24
Pedersen, Jan	60.00	3.75	15	24
Sibun, Penelope	60.00	3.75	15	24
Hanks, Patrick	49.50	5.75	18	15
Papineni, Kishore	43.29	4.99	19	18
Smadja, Frank	35.50	1.56	20	54
Remaining authors in top 20 by S-AIF				
Carletta, Jean	24.20	12.15	26	7
Vaithyanathan, Shivakumar	33.00	9.13	24	8
Turney, Peter	34.33	7.92	21	12
Pang, Bo	22.00	6.19	28	14
Popat, Ashok C.	21.00	5.38	29	16
Dean, Jeffrey	21.00	5.38	29	16
Goldstein, Jade	8.00	4.38	52	19
Kantrowitz, Mark	8.00	4.38	52	19

5.2.2. Sentiment-based h-index (S_h-index)

The h-index [13] serves as a metric to assess the scholarly influence of an author, encompassing both the volume and significance of their publications. It signifies the count of articles (h) that have accrued no less than h citations, serving as an indicator of acknowledgment and impact. The computation methodology for h-index is outlined in Algorithm 1, providing a clear depiction of its computation process.

Algorithm 1 Computation of the h-index

```
Require: NP: Number of Publications, Citations: Array of publication's citations, sorted
  in descending order
  $h \leftarrow 0$, $index \leftarrow 1$
  while $index \leq$ NP do
      if Citations[index] $\geq$ index then
          $h \leftarrow index$
          $index \leftarrow index + 1$
      else
          break
      end if
  end while
  Result: h-index $\leftarrow h$
```

While the h-index enjoys widespread usage, it primarily relies on quantitative factors such as the count of citations. We introduce an innovative sentiment-driven alternative metric. In contrast to the method proposed by Zeng Ma [21], which suggests a modified version of the traditional h-index for authors using a specific equation, our approach centers on evaluating the collective sentiment expressed within citations for each article.

The proposed metric involves examining the articles(s) published by a particular author, arranged in descending order based on their "total sentiment score". This examination provides a meaningful gauge of an author's work credibility and impact. An author is said to have a sentiment-based h-index (called S_h-index) score of 's' when each of the top 's' articles, ranked by total sentiment score, receives a score of at least 's'. The computational procedure for deriving the S_h-index is elaborated upon in Algorithm 2.

Algorithm 2 Calculation of the sentiment-based h-index (S_h-index)

```
Require: NP: Number of Publications, Sentiment: Array of total sentiment score for pub-
  lications, sorted in descending order
  $s \leftarrow 0$, $index \leftarrow 1$
  while $index \leq$ NP do
      if Sentiment [index] $\geq$ index then
          $s \leftarrow index$
          $index \leftarrow index + 1$
      else
          break
      end if
  end while
  Result: $S_h$-index $\leftarrow s$
```

Table 7 shows the author rankings based on h-index and S_h-index for the top 20 authors.

Table 7
Top 20 authors (Corpus 1) by h-index

Author	h-index	S_h-index	Rank by	
			h-index	S_h-index
Klein, Dan	6	1	1	30
Marcu, Daniel	5	3	2	1
Collins, Michael	5	3	2	1
Della Pietra, Vincent J.	4	3	4	1
Lee, Lillian	4	3	4	1
Manning, Christopher	4	1	4	30
Turney, Peter	3	2	7	5
Gildea, Daniel	3	1	7	30
Och, Franz Josef	3	2	7	5
McDonald, Ryan	3	2	7	5
Johnson, Mark	3	2	7	5
Blitzer, John	3	2	7	5
Pereira, Fernando	3	2	7	5
Hovy, Eduard	3	1	7	30
Ng, Andrew	3	2	7	5
Daume III, Hal	3	2	7	5
Brown, Peter F.	3	2	7	5
Della Pietra, Stephen	3	2	7	5
Mercer, Robert L.	3	2	7	5
Knight, Kevin	3	2	7	5
Remaining articles in top 20 by S_h-index				
Wu, Dekai	2	2	22	5
Smith, David	2	2	22	5
Brants, Thorsten	2	2	22	5

5.2.3. Metric based on PageRank and sentiment score (S-PageRank)

PageRank [8] is a graph-centered ranking algorithm that evaluates the significance of a vertex in a graph by factoring in its incoming and outgoing links [28]. In the field of author ranking, using the author citation network, a PageRank-based citation metric can gauge the author's prestige.

In the present work, we investigate author evaluation by combining the sentiment score with the PageRank approach. We present an evaluation of authors' prestige based on citation frequency vs. the sentiment score of citations.

PageRank was proposed by Google for its search engine to rank websites by considering the quantitative (frequency) and qualitative aspects of web links. The idea was to determine higher prestige web pages. The original algorithm proposed a damping factor, a value between 0 and 1, as the likelihood of a web link on the current page being picked. The choice of damping factor significantly impacts the behavior of the PageRank algorithm, prompting a closer investigation into its consequences.

Importantly, previous research [7] indicates that the optimal damping factor range varies depending on the intended purpose. Notably, the range of 0.8 to 0.9 aligns with the concept of "Weak Rank". This alignment proves valuable in scenarios prioritizing relevant results, such as web search engines. In such cases, avoiding false negatives is more important than preventing false positives. In the web search example, a false negative being a relevant page not being ranked and a false positive being an irrelevant page being ranked.

Conversely, the emphasis on "Strong Rank", achieved with damping factors between 0.5 and 0.6, is advantageous in trust and reputation systems. These systems aim to minimize false positives, making them well-suited for evaluating trustworthy items where it is crucial to avoid ranking untrustworthy items highly [7]. For instance, in academic contexts, it's particularly important to avoid false positives (untrustworthy articles being ranked highly). The aforementioned insights suggest that it is advisable to prioritize PageRank with a damping factor of 0.55 in the field of scholarly articles, given its strong reliance on trust and reputation.

We empirically tested and analyzed the rankings generated for damping factor (d) values of 0.85 and 0.55. For d = 0.85, change in rankings for a few authors appeared harder to justify. For instance, upon analyzing the author "Carletta, Jean," it was observed that she was cited in a total of 124 articles and her overall sentiment score amounted to 60. Overall, she received 74 positive citations, 21 negative citations, and the rest were neutral. Citation frequency-based PageRank placed her at rank 11. Surprisingly, despite a majority of positive citations, her rank by sentiment-PageRank positioned her notably lower at 3044. In contrast, when employing a damping factor of 0.55, these ranks adjusted to 9 and 30, respectively.

It is noteworthy that Ghosh and Shah [12] delved into PageRank-based indices for articles but lacked mention of the damping factor, a critical parameter explored by us across two distinct values in our research. Additionally, we are investigating the PageRank-based metric for authors.

The results presented in Table 8 highlight the top 20 authors, ranked based on both PageRank score and on Sentiment-based PageRank (S-PageRank) score.

Table 8

Top 20 authors (Corpus 1) by PageRank

Author	PageRank $(\cdot 10^{-2})$	S-PageRank $(\cdot 10^{-2})$	Rank By	
			PageRank	S-PageRank
Della Pietra, Vincent J.	2.73	3.42	1	1
Della Pietra, Stephen	2.29	2.52	2	4
Marcus, Mitch	2.13	1.87	3	5
Collins, Michael	1.87	0.918	4	15
Brown, Peter F.	1.76	2.87	5	2
Mercer, Robert L.	1.76	2.87	5	2
Santorini, Beatrice	1.59	1.31	7	9
Marcinkiewicz, Mary Ann	1.59	1.31	7	9

Table 8 cont.

Author	PageRank $(\cdot 10^{-2})$	S-PageRank $(\dot{1}0^{-2})$	Rank By	
			PageRank	S-PageRank
Carletta, Jean	1.43	0.418	9	30
Yarowsky, David	1.34	1.49	10	7
Och, Franz Josef	1.26	1.31	11	8
Ratnaparkhi, Adwait	1.16	1.02	12	12
Dunning, Ted	1.08	1.14	13	11
Berger, Adam L.	0.989	0.576	14	25
Lee, Lillian	0.781	0.141	15	52
Turney, Peter	0.744	−0.209	16	3033
Roukos, S.	0.727	0.895	17	16
Papineni, Kishore	0.727	0.895	17	16
Ward, Todd	0.727	0.895	17	16
Zhu, Wei-Jing	0.727	0.895	17	16
Remaining articles in top 20 by S-PageRank				
Koehn, Philipp	0.548	1.55	26	6
Lai, Jenifer C.	0.480	0.942	29	13
deSouza, Peter V.	0.464	0.924	31	14

5.3. Objective 3: evaluation of the proposed metrics for ranking authors

Our method for assessing the proposed sentiment-based metrics involves comparing them with their quantitative counterparts. As each indexing approach yields a unique sequence of articles in the collection, it's crucial to determine whether substantial divergences exist between these ordered lists. We employ two approaches to measure the degree of similarity or difference between two ranked lists:

1. Kendall's Tau Rank Correlation,
2. Rank Biased Distance (RBD).

5.3.1. Kendall's tau rank correlation

Kendall's Tau Rank Correlation evaluates the degree of association or correlation between two sets of data by measuring the probability that two items retain their order in both lists. Positive values indicate a direct monotonic relationship, while negative values suggest an inverse relationship. Correlation coefficients with absolute values greater than 0.7 signify a strong level of correlation between variables, between 0.5 to 0.7 suggest a moderate level of correlation, and absolute values less than 0.5 indicate a weak correlation. The p-value remains a pivotal indicator of the statistical significance of these correlations, enabling us to evaluate the robustness of our findings. One remarkable feature of Kendall's Tau is its ability to handle ties effectively as compared to Spearman's Correlation Coefficient [2].

When analyzing the results for Corpus 1, we observe that Kendall's Tau (τ) values consistently lie within the range of 0.35 to 0.5 for variables such as Article/Author

Number of Citations & Sentiment Scores, AIF & S-AIF, h-index & S_h-index. This range signifies a weak monotonic association and a significant deviation in the author rankings for these proposed metrics when compared to their respective quantitative counterparts. However, in the case of Author PageRank (d = 0.55), Kendall's Tau (τ) coefficient is relatively higher at 0.588, indicating a moderately strong relationship. It's worth noting that the findings are statistically significant at the 1% significance level.

In the context of Corpus 2, it becomes evident that Kendall's Tau (τ) for all cases except the PageRank-based approach shows a weakly positive correlation. Although Kendall's Tau Coefficient for AIF & S-AIF and H & S_h-index remain statistically significant, the p-values for Author and Article Scores stand at 0.38 and 0.73, respectively, suggesting a lack of reliability in these cases.

In summary, sentiment awareness causes a higher disruption in frequency-based metrics as compared to PageRank-based metrics for author ranking. Table 9 displays the computed Kendall's coefficients for the proposed indices in comparison to their quantitative counterparts for both datasets.

Table 9

Kendall's Tau, p-value, and Rank Biased Distance (RBD) for the proposed sentiment-based metrics compared to their quantitative counterparts

Metrics	Corpus 1			Corpus 2		
	Kendall (τ)	*p*-value	RBD	Kendall (τ)	*p*-value	RBD
Article's Number of Citations & Sentiment Score	0.419	$1.19 \cdot 10^{-15}$	0.385	-0.010	0.408	0.999
Author's Number of Citations & Sentiment Score	0.433	$1.26 \cdot 10^{-26}$	0.601	0.002	0.74	0.999
AIF & S-AIF	0.397	$1.064 \cdot 10^{-26}$	0.370	0.014	0.0089	0.999
h-index & S_h-index	0.494	$5.42 \cdot 10^{-20}$	0.314	0.017	0.0094	0.913
PageRank & S-PageRank (d = 0.55)	0.588	$2.03 \cdot 10^{-253}$	0.285	0.795	0.00	0.282

To understand the above results better, we studied the characteristics of both corpora. Diverging from the concentrated citation pattern observed in Corpus 1 (with a total of 195 cited articles), Corpus 2 exhibits a distinctive arrangement. An intriguing observation emerges within Corpus 2 that a substantial 6,405 articles, out of 11,020 instances, are referenced by just a single article each. This distribution highlights that a notable proportion of articles in Corpus 2 have garnered minimal citations. Consequently, these articles tend to be clustered together in terms of rank, a consequence of their shared attribute of possessing only one citation.

Equally noteworthy, this trend extends to the realm of authors in Corpus 2. Among the 24,024 cited authors, a significant 22,890 authors also have just single citations. As a result, these instances end up occupying identical ranks in relation to the number of citing papers.

The sentiment score introduces an element of diversity due to its reliance on the content of the citation sentence, rather than being influenced solely by the quantity of citations. Notably, these metrics serve as effective differentiators, functioning as tiebreakers for example in our case where a large number of authors of the corpus exhibits only one citation, as is further shown in Section 6.

5.3.2. Rank biased distance (RBD)

Rank Biased Distance (RBD) [35] is a metric to measure the extent of deviation between the two ranked lists, giving more weight to the elements at higher ranks as compared to those at the lower ranks. While Kendall's Tau rank correlation coefficient does not distinguish between ranks based on position as long as the relative order is preserved, in the context of ranked lists, it is often considered that items ranked higher carry greater importance than those ranked lower. In other words, a rank switch at the top of a ranked list should lower the similarity score compared to a rank switch at the bottom of the list. RBD is sensitive to the switching of the ranks of articles at the top of the ranked list and is computed using the Equation (5) presented below:

$$RBD_k = (1-p)\sum_{d=1}^{k} \frac{|S_d \ \Delta \ T_d|}{2d} \cdot p^{d-1} \tag{5}$$

In the formula for RBD presented above, S and T are the two ranked lists being compared, each containing k elements. Δ symbolizes the deviation between the two ranked lists. p is a weighing factor with a value in the range $[0, 1]$. A lower value of p assigns more weight to higher-ranked elements. Since a higher value of p evenly distributes the weight among all the elements of the ranked list, a value of 0.9 was adopted for p, which is also in line with recommendations from the original work [35]. Next, we interpret the Rank Biased Distance (RBD) results, tabulated in Table 9.

In Corpus 1, for most indices, the RBD value is lower than 0.4, suggesting substantial but reasonable ranking shifts when comparing sentiment-based and citation-based metrics. Notably, the RBD value for the Author's Number of Citations vs. Sentiment Score exhibits a relatively higher deviation of 0.601, signifying significant disparities.

In contrast, in Corpus 2, citation-based rankings tend to cluster around specific values, while sentiment-based rankings exhibit broader variation based on sentiment scores. This variability likely contributes to the increased deviations for most indices with RBD scores close to 1. Additionally, the PageRank-based index in Corpus 2 shows a lower deviation since it already has an embedded consideration of prestige.

6. Discussion

6.1. Discussion on results for objective 1

Our first objective dealt with the computation of aggregate sentiment scores for authors and its experimental results are tabulated in Tables 4 and 5. To better explain the results, we analyzed the article "P03-1021" (Och and Ney, 2003) (Tab. 4) that exhibited maximum rank deviation. It is observed that many of the citation sentences point out the limitations of the article "P03-1021" and highlight the ongoing advancements in the field. Following are two such examples of citation sentences:

"The most widely applied training procedure for statistical machine translation IBM model 4 (Brown et al., 1993) unsupervised training followed by post-processing with symmetrization heuristics (Och and Ney, 2003) yields low-quality word alignments."

"Unlike minimum error rate training (Och and Ney, 2003), our system can exploit large numbers of specific features in the same manner as static reranking systems (Shen et al., 2004; Och et al., 2004)."

The above citation sentences express a negative sentiment for the article "P03-1021" and talk about the progressions beyond it, thereby validating the rank deviation.

In addition, we consider the following citation sentence citing the article "W05-0904" (Liu and Gildea, 2005). The provided sentence exemplifies a scenario in which criticism is succeeded by a subtle acknowledgment of merit.

"Our method, extending this line of research with the use of labelled LFG dependencies, partial matching, and n-best parses, allows us to considerably outperform Liu and Gildea's (2005) highest correlations with human judgement (they report 0.144 for the correlation with human fluency judgement, 0.202 for the correlation with human overall judgement), although it has to be kept in mind that such comparison is only tentative, as their correlation is calculated on a different test set."

The above sentence conveys a negative opinion about the cited article "W05-0904." However, the tone of the sentence appears positive in that it praises the citing article's work. Analyzing the sentiment score, the given sentence has an overall sentiment score of "-0.75". Meanwhile, the citation frequency-based and sentiment score-based evaluations resulted in 48 and 151, respectively, indicating that the criticism is noted, despite the positive tone of the sentence, while manifesting the cited article's rank.

6.2. Discussion on results for objective 2

In the present section, we discuss the results for each of the three developed sentiment-enhanced citation metrics, namely: S-AIF: Sentiment-based Author Impact Factor, S_h-index: Sentiment-based h-index, and S-PageRank: Sentiment-based PageRank.

6.2.1. S-AIF

The results for S-AIF are tabulated in Table 6. An analysis of the rankings given in Table 6 emphasizes the interaction between citation-based and sentiment-based impact by highlighting cases in which AIF and S-AIF rankings don't demonstrate substantial changes and situations in which they have significant variations.

As an illustration, writers such as "Santorini, Beatrice" and "Marcinkiewicz, Mary Ann" (Tab. 6) maintain a similar position in both metrics, however "Smadja, Frank" and "Mercer, Robert L." possess a substantially higher rank by AIF (20 & 9) compared to their rank by S-AIF (54 & 38).

Analyzing the author "Santorini, Beatrice" (Tab. 6), we discovered that they have only one article titled "J93-2004." This article has been cited in 434 other articles, with a total of 626 citation sentences. Among these sentences, 264 received a sentiment score of "0," 246 received a score higher than 0, and 116 received a negative sentiment score. The prevalence of neutral and positive sentences has resulted in a high S-AIF score of 62.13. The high AIF as well as the high S-AIF indicates indicates that the author's work has received an overall positive recognition, resulting in a strong ranking in both metrics.

Examining the author "Mercer, Robert L." (Tab. 6) with an AIF score of 122.50, we found that he has authored four publications. These articles have been cited in 490 other articles, generating 918 citation sentences. Out of these sentences, 408 received a sentiment score of "0," 270 had a score above 0, and 240 had a negative sentiment score. Although there are more neutral sentences, the number of positive and negative sentences is roughly balanced, resulting in a slightly positive overall sentiment score of 2.16. The presence of a slightly positive sentiment score as compared to a moderately strong AIF score explains the observed deviation. This suggests that Robert's work might have garnered a significant number of citations, contributing to a high AIF rank, but the sentiment analysis reveals a lower emotional impact associated with their work.

6.2.2. S_h-index

Table 7 shows the results for S_h-index. It can be observed that "Marcu, Daniel" and "Collins, Michael" (Tab. 7) maintain stable positions across both metrics, while "Klein, Dan" and "Manning, Christopher" hold notably higher h-index ranks (1 & 4) compared to their S_h-index ranks (30).

Examining all seven articles authored by "Klein, Dan" (Tab. 7), we observed that six of these articles were cited by six or more other articles, resulting in an h-index of 6. However, when evaluating sentiment scores, only four out of the seven articles received an overall positive score, with just one article achieving a score greater than 1. Consequently, the S_h-index was determined to be one. That is, the works of "Klein, Dan" garnered citations, boosting his h-index, but also received lower sentiment scores, suggesting a less intense emotional impact associated with his articles.

In summary, Table 7 highlights the interplay between citation-based influence and sentiment-based h-index rankings. It showcases scenarios with consistent h-index and S_h-index rankings as well as those with significant disparities.

6.2.3. S-PageRank

The results for S-PageRank are presented in Table 8. Upon analyzing the data, it becomes evident that certain authors, such as "Della Pietra, Vincent J.", "Della Pietra, Stephen", and "Marcus, Mitch" (Tab. 8) exhibit minimal disparity between these two ranking methods. Conversely, the author "Turney, Peter" demonstrates a substantial discrepancy of 3017 ranks between the two rankings.

Furthermore, within the context of the PageRank algorithm, it becomes apparent that authors lacking any citations within the considered dataset would receive a popularity score of zero for conventional metrics like the h-index and impact factor. However, the PageRank-based metrics could potentially assign a non-zero score to such authors resulting in interesting results such as an uncited author ranked higher than a cited author.

We consider an uncited author "Liu, Hao" who received a PageRank score of 0.00018 and an S-PageRank score of 0.00021. On the other hand, an author "Klein, Dan" has been cited in 86 articles with a PageRank score of 0.00372 and an S-PageRank score of -0.00995. Upon analyzing "Klein, Dan," we found that 53 authors cited him negatively, 51 cited him positively, and 36 cited him with a sentiment score of 0 resulting in him receiving a lower S-PageRank based rank than the uncited author.

6.3. Discussion on results for objective 3

The evaluation of the proposed metrics uncovers valuable insights (Tab. 9). In Corpus 1, Kendall's coefficient reveals a weak and positive link between citation-based metrics and the proposed sentiment-based metrics. Notably, this relationship triggers substantial shifts in author rankings for articles with pronounced (positive or negative) sentiment. For instance, consider the author "Banerjee, Satanjeev" with 55 citations from 3 publications and a cumulative sentiment score of -1.875 showing an overall negative sentiment of the citations. His AIF score of 18.33 contrasts sharply with an S-AIF of -0.625, leading to a significant rank difference of 262 between AIF (Rank 32) and S-AIF (Rank 294). A similar trend of high deviation is observed for h-index vs. S_h-index (rank difference of 167) and S-PageRank and PageRank (rank difference of 2993). These variations underscore the significant impact of (negative) sentiment on metrics for "Banerjee, Satanjeev".

In Corpus 2, Kendall's coefficient for PageRank vs. S-PageRank exhibited a robust correlation of approximately 0.82. AIF/S-AIF and h-index/S_h-index yielded weak but positive and significant correlation.

Notably, corpus 2 contains a large number of articles and authors with the same rank values for quantitative metrics. The data presented in Table 10 illustrates four distinct instances from corpus 2, wherein authors possess a citation count of one. Although the citation count results in an Author Impact Factor (AIF) of 1 for each case, the varying sentiment scores of the associated articles contribute to divergent Sentiment-based Author Impact Factors (S-AIF). Consequently, while the AIF rankings position all instances similarly, the S-AIF yields disparate results, prioritizing qualitative factors over frequency-based parameters. This nuanced approach proves valuable, particularly for recent research articles, by effectively distinguishing superior quality papers within the academic landscape.

Table 10

Authors with a single citation but different sentiment scores.
While their AIF-based rank is the same, the S-AIF-based ranking shows a marked difference

Cited Author's ID	Citation Text	Sentiment Score	Ranks by	
			AIF	S-AIF
1802065	Eyelid suture, on the other hand, produces a novel pattern of thin dark columns alternating with wide pale columns (Horton, 1984; Hendry and Jones, 1986; Crawford et al., 1989; Trusk et al., 1990, Tigges et al., 1992)	0.5	11588	4414.5
3881197	CDDO-Me is a synthetic triterpenoid that was under phase III clinical development for the treatment of advanced chronic kidney disease (37, 38) However, due to adverse events in the phase III clinical trial, further development of CDDO-Me was terminated (39)	−1.5	11588	23267
10616537	Our data also showed that lesions in the head and neck region responded better to PDL therapy than in other regions; similar results were reported by several studies [16, 24, 28]	0.75	11588	2681
145396957	The recent identification of lymphatic endothelial-specific markers, such as hyaluronic acid receptor-1 (LYVE-1) [2], has greatly increased attention on how lymphangiogenesis, the growth of lymphatic vessels, is regulated in the tumor microenvironment	0.125	11588	8992.5

For Corpus 1, it is observed that the inclusion of sentiment in evaluating the impact of authors and articles leads to at least a 28% difference between the ranking metrics. For Corpus 2, this difference is about 99% in most of the metrics but for PageRank-based metrics, the difference is the minimum at 28%.

PageRank exhibited higher Kendall's coefficients in both Corpus 1 and Corpus 2 because it incorporates prestige when assessing node importance within a network. Consequently, the correlation between PageRank and S-PageRank remained consistently strong, indicating that the addition of sentiment enhances results to a limited extent, preserving the core prestige-based attributes of PageRank. This consideration differentiates it from metrics solely focused on popularity.

7. Summary, conclusions, and future work

In this article, we examined how sentiments influence both, individual articles and their authors. Our article's overall focus can be categorized into four main sections. Our initial step was to evaluate the sentiment score of each citation sentence. To accomplish this, we applied text preprocessing techniques and subsequently harnessed the SentiWordNet lexical resource for calculating the sentiment score of each citation sentence. Through this process, we derived sentiment scores for each instance within the Citation Sentiment Corpus [5] (Corpus 1) and Citation Intent Classification Dataset [9] (Corpus 2).

Our second task involved the computation of overall sentiment scores for both articles and authors. This involved a multi-step process. Initially, we extracted author information for each article. Subsequently, we implemented a self-citation filter by identifying pairs of citing and cited articles that shared at least one author in common. Finally, we aggregated the sentiment scores associated with each citation within an article to determine the “Total Sentiment Score” for that specific article. Similarly, for author sentiment scores, we aggregated the “Total Sentiment Scores” derived from all articles authored by that individual.

Our third task was to introduce sentiment-driven metrics to evaluate authors and articles, leveraging the total sentiment scores computed in the previous step. Building upon commonly used quantitative metrics like Author Impact Factor (AIF), h-index, and PageRank, we put forth their sentiment-oriented equivalents, namely S-AIF, S_h-index, and S-PageRank.

Finally, we assessed the performance of the proposed sentiment-driven indices in comparison to their quantitative counterparts, using Kendall's Tau correlation coefficients and associated p-values. Along with this, we also employed Rank Biased Distance (RBD) to quantify the deviation between the order of rankings.

In both datasets, for most of the metrics, a weak but monotonic relationship was observed between the frequency-based and sentiment-based metrics. Interestingly, across both corpora, the Pagerank-based indices exhibited a higher correlation, indicating that prestige-based ranking had a stronger association between citation-based and sentiment-based counterparts as compared to purely popularity-based ranking. The present study also demonstrated that a sentiment-aware index leads to at least 28% rank-based deviation in author ranks from a citation-based index.

In summary, we addressed all our research objectives successfully. We established a method for computing a composite score value for an author. The proposed sentiment-infused citation metrics revealed significant shifts in author rankings for articles that were cited with a stronger (positive or negative) sentiment.

This relationship triggers substantial shifts in author rankings for articles with pronounced (positive or negative) sentiment.

Our work establishes the importance of considering the qualitative aspects of a citation while evaluating the influence of an authors' work. An evaluation of scholarly impact solely on the basis of citation frequency tends to follow the tenet of "rich becoming richer". Incorporating sentiments in the analysis of an author's impact lends more depth to the evaluation. However, we believe that more research needs to be carried out on larger data to establish the widespread use of sentiment-aware indices.

The field of Scientometrics is poised to consider the sentiments of citations during the evaluation of authors and their articles. To do the same, the researchers require a shift in the way they analyze the research output. However, to achieve this, it is necessary to make the computation and use of the proposed sentiment-based metrics easy and accessible.

Our analysis is focused on determining the significance of an article from the eye of the sentiments expressed by the citing authors. Notably, certain aspects of context and period can be reflected by the sentiment embedded in the citation. For instance, certain aged publications amass a substantial number of citations, maintaining a prominent position in scholarly rankings despite their outdated nature. While these outdated articles may initially garner positive sentiments, over time, they transition into a reference in the literature, eliciting predominantly neutral sentiments. By considering sentiments, our approach aims to provide a more nuanced evaluation, allowing newer and impactful articles of authors a chance to gain better ranking positions.

In the future, we may study the integration of factors such as the citation distribution of an article/author, and the proportion of neutral citations into the ranking of scholarly articles and authors. We also plan to develop an automated system that given an author's details should extract citation sentences from articles citing the authors' publications and compute the author's sentiment score and the value of the sentiment-based metrics.

References

[1] Abu-Jbara A., Ezra J., Radev D.: Purpose and polarity of citation: Towards NLP-based bibliometrics. In: *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 596–606, 2013.

[2] Akoglu H.: User's guide to correlation coefficients, *Turkish Journal of Emergency Medicine*, vol. 18(3), pp. 91–93, 2018. doi: 10.1016/j.tjem.2018.08.001.

[3] Aljuaid H., Iftikhar R., Ahmad S., Asif M., Tanvir Afzal M.: Important citation identification using sentiment analysis of in-text citations, *Telematics and Informatics*, vol. 56, 101492, 2021. doi: 10.1016/j.tele.2020.101492.

[4] Amjad Z., Ihsan I.: VerbNet based Citation Sentiment Class Assignment using Machine Learning, *International Journal of Advanced Computer Science & Applications*, vol. 11(9), 2020. doi: 10.14569/ijacsa.2020.0110973.

[5] Athar A.: Sentiment analysis of citations using sentence structure-based features. In: S. Petrovic, E. Selfridge, E. Pitler, M. Osborne, T. Solorio (eds.), *Proceedings of the ACL 2011 Student Session*, pp. 81–87, Association for Computational Linguistics, Portland, OR, USA, 2011. https://aclanthology.org/P11-3015/.

[6] Baccianella S., Esuli A., Sebastiani F.: SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, D. Tapias (eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pp. 2200–2204, European Language Resources Association (ELRA), Valletta, Malta, 2010. http://lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf.

[7] Bressan M., Peserico E.: Choose the damping, choose the ranking?, *Journal of Discrete Algorithms*, vol. 8(2), pp. 199–213, 2010. doi: 10.1016/j.jda.2009.11.001.

[8] Brin S., Page L.: The anatomy of a large-scale hypertextual web search engine, *Computer networks and ISDN systems*, vol. 30(1-7), pp. 107–117, 1998. doi: 10.1016/s0169-7552(98)00110-x.

[9] Cohan A., Ammar W., Zuylen M.V., Cady F.: Structural Scaffolds for Citation Intent Classification in Scientific Publications. In: J. Burstein, C. Doran, T. Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019. doi: 10.18653/v1/N19-1361.

[10] Fellbaum C.: WordNet. In: R. Poli, M. Healy, A. Kameas (eds.), *Theory and applications of ontology: computer applications*, pp. 231–243, Springer, Dordrecht, 2010. doi: 10.1007/978-90-481-8847-5_10.

[11] Garfield E.: Journal impact factor: a brief review, *CMAJ: Canadian Medical Association Journal,*, vol. 161(8), pp. 979–980, 1999. https://www.cmaj.ca/content/161/8/979.short.

[12] Ghosh S., Shah C.: Identifying Citation Sentiment and its Influence while Indexing Scientific Papers. In: *Proceedings of the 53rd Hawaii International Conference on System Sciences*, pp. 1–10, 2020. doi: 10.24251/hicss.2020.307.

[13] Hirsch J.E.: An index to quantify an individual's scientific research output, *Proceedings of the National Academy of Sciences*, vol. 102(46), pp. 16569–16572, 2005. doi: 10.1073/pnas.0507655102.

[14] Ihsan I., Qadir M.A.: CCRO: Citation's Context & Reasons Ontology, *IEEE Access*, vol. 7, pp. 30423–30436, 2019. doi: 10.1109/access.2019.2903450.

[15] Karim M., Missen M.M.S., Umer M., Sadiq S., Mohamed A., Ashraf I.: Citation Context Analysis Using Combined Feature Embedding and Deep Convolutional Neural Network Model, *Applied Sciences*, vol. 12(6), 3203, 2022. doi: 10.3390/app12063203.

[16] Kazi P., Patwardhan M., Joglekar P.: Towards a new perspective on context based citation index of research articles, *Scientometrics*, vol. 107, pp. 103–121, 2016. doi: 10.1007/s11192-016-1844-2.

[17] Kilicoglu H., Peng Z., Tafreshi S., Tran T., Rosemblat G., Schneider J.: Confirm or refute?: A comparative study on citation sentiment classification in clinical research publications, *Journal of Biomedical Informatics*, vol. 91, 103123, 2019. doi: 10.1016/j.jbi.2019.103123.

[18] Kinney R.M., Anastasiades C., Authur R., Beltagy I., Bragg J., Buraczynski A., Cachola I., *et al.*: The Semantic Scholar Open Data Platform, *ArXiv*, 2023. doi: 10.48550/arXiv.2301.10140.

[19] Kochhar S.K., Ojha U.: Index for objective measurement of a research paper based on sentiment analysis, *ICT Express*, vol. 6(3), pp. 253–257, 2020. doi: 10.1016/j.icte.2020.02.001.

[20] Liu H.: Sentiment analysis of citations using word2vec, *arXiv preprint arXiv:170400177*, 2017. doi: 10.48550/arXiv.1704.00177.

[21] Ma Z., Nam J., Weihe K.: Improve sentiment analysis of citations with author modelling. In: A. Balahur, E. van der Goot, P. Vossen, A. Montoyo (eds.), *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 122–127, Association for Computational Linguistics, San Diego, California, 2016. doi: 10.18653/v1/w16-0420.

[22] MacRoberts M.H., MacRoberts B.R.: The negational reference: Or the art of dissembling, *Social Studies of Science*, vol. 14(1), pp. 91–94, 1984. doi: 10.1177/030631284014001006.

[23] Malkawi R., Daradkeh M., El-Hassan A., Petrov P.: A Semantic Similarity-Based Identification Method for Implicit Citation Functions and Sentiments Information, *Information*, vol. 13(11), 546, 2022. doi: 10.3390/info13110546.

[24] Mäntylä M.V., Graziotin D., Kuutila M.: The evolution of sentiment analysis – A review of research topics, venues, and top cited papers, *Computer Science Review*, vol. 27, pp. 16–32, 2018. doi: 10.1016/j.cosrev.2017.10.002.

[25] Mercier D., Bhardwaj A., Dengel A., Ahmed S.: SentiCite: An Approach for Publication Sentiment Analysis, *arXivorg*, 2019. doi: 10.48550/arXiv.1910.03498.

[26] Nazir S., Asif M., Ahmad S., Aljuaid H., Iftikhar R., Nawaz Z., Ghadi Y.Y.: Important Citation Identification by Exploding the Sentiment Analysis and Section-Wise In-Text Citation Weights, *IEEE Access*, vol. 10, pp. 87990–88000, 2022. doi: 10.1109/access.2022.3199420.

[27] Raza H., Faizan M., Hamza A., Mushtaq A., Akhtar N.: Scientific Text Sentiment Analysis using Machine Learning Techniques, *International Journal of Advanced Computer Science & Applications*, vol. 10(12), 2019. doi: 10.14569/ijacsa.2019.0101222.

[28] Sardar A., Pramanik P.K.D.: Estimating Authors' Research Impact Using PageRank Algorithm. In: *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2021, Volume 2*, pp. 471–483, Springer, 2022. doi: 10.1007/978-981-16-2937-2_29.

[29] Teufel S., Siddharthan A., Tidhar D.: Automatic classification of citation function. In: *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 103–110, 2006. doi: 10.3115/1610075.1610091.

[30] Umer M., Sadiq S., Missen M.M.S., Hameed Z., Aslam Z., Siddique M.A., Nappi M.: Scientific papers citation analysis using textual features and SMOTE resampling techniques, *Pattern Recognition Letters*, vol. 150, pp. 250–257, 2021. doi: 10.1016/j.patrec.2021.07.009.

[31] Umer M., Sadiq S., Missen M.M.S., Hameed Z., Aslam Z., Siddique M.A., Nappi M.: Scientific papers citation analysis using textual features and SMOTE resampling techniques, *Pattern Recognition Letters*, vol. 150, pp. 250–257, 2021. doi: 10.1016/j.patrec.2021.07.009.

[32] Valenzuela M., Ha V., Etzioni O.: Identifying Meaningful Citations. In: *Proceedings of AAAI Workshop Papers 2015, Scholarly Big Data: AI Perspectives, Challenges, and Ideas*, vol. 15, pp. 21–26, 2015. https://cdn.aaai.org/ocs/ws/ws0121/10185-46008-1-PB.pdf.

[33] Visser R., Dunaiski M.: Sentiment and intent classification of in-text citations using BERT. In: *Proceedings of 43rd Conference of the South African Institute of Computer Scientists and Information Technologists*, vol. 85, pp. 129–145, 2022. doi: 10.29007/wk21.

[34] Wang M., Zhang J., Jiao S., Zhang X., Zhu N., Chen G.: Important citation identification by exploiting the syntactic and contextual information of citations, *Scientometrics*, vol. 125, pp. 2109–2129, 2020. doi: 10.1007/s11192-020-03677-1.

[35] Webber W., Moffat A., Zobel J.: A similarity measure for indefinite rankings, *ACM Transactions on Information Systems (TOIS)*, vol. 28(4), pp. 1–38, 2010. doi: 10.1145/1852102.1852106.

[36] Xu L., Ding K., Lin Y., Zhang C.: Does citation polarity help evaluate the quality of academic papers?, *Scientometrics*, pp. 4065–4087, 2023. doi: 10.1007/s11192-023-04734-1.

[37] Yan E., Chen Z., Li K.: Authors' status and the perceived quality of their work: Measuring citation sentiment change in nobel articles, *Journal of the Association for Information Science and Technology*, vol. 71(3), pp. 314–324, 2020. doi: 10.1002/asi.24237.

[38] Yousif A., Niu Z., Chambua J., Khan Z.Y.: Multi-task learning model based on recurrent convolutional neural networks for citation sentiment and purpose classification, *Neurocomputing (Amsterdam)*, vol. 335, pp. 195–205, 2019. doi: 10.1016/j.neucom.2019.01.021.

[39] Yousif A., Niu Z., Tarus J.K., Ahmad A.: A survey on sentiment analysis of scientific citations, *Artificial Intelligence Review*, vol. 52, pp. 1805–1838, 2019. doi: 10.1007/s10462-017-9597-8.

Affiliations

Shikha Gupta
Shaheed Sukhdev College of Business Studies, New Delhi, India, shikhagupta@sscbsdu.ac.in

Animesh Kumar
Shaheed Sukhdev College of Business Studies, New Delhi, India,
animesh.21508@sscbs.du.ac.in

Received: 06.01.2024
Revised: 03.04.2024
Accepted: 03.04.2024

Chiheb Eddine Ben Ncir
Mohamed Aymen Ben Haj Kacem
Mohammed Alattas

EXPLAINABLE SPARK-BASED PSO CLUSTERING FOR INTRUSION DETECTION

Abstract *Given the exponential growth of available data in large networks, the existence of rapid, transparent, and explainable intrusion detection systems has become of highly necessity to effectively discover attacks in such huge networks. To deal with this challenge, we propose a novel explainable intrusion detection system based on Spark, Particle Swarm Optimization (PSO) clustering, and eXplainable Artificial Intelligence (XAI) techniques. Spark is used as a parallel processing model for the effective processing of large-scale data, PSO is integrated to improve the quality of the intrusion detection system by avoiding sensitive initialization and premature convergence of the clustering algorithm and finally, XAI techniques are used to enhance interpretability and explainability of intrusion recommendations by providing both micro and macro explanations of detected intrusions. Experiments are conducted on large collections of real datasets to show the effectiveness of the proposed intrusion detection system in terms of explainability, scalability, and accuracy. The proposed system has shown high transparency in assisting security experts and decision-makers to understand and interpret attack behavior.*

Keywords Intrusion Detection System (IDS), Artificial Intelligence (AI), Explainable AI (XAI), Particle Swarm Optimization (PSO), Spark framework

Citation Computer Science 25(2) 2024: 211–237

1. Introduction

Emerging information technologies, such as cloud computing and control systems, have undergone rapid evolution in recent years. These systems often handle large volumes of information across diverse infrastructures and complex networks. Consequently, numerous malicious entities attempt to compromise these systems by exploiting communication networks. To secure computer systems and networks from unauthorized access and data breaches, Intrusion Detection Systems (IDS) are implemented. IDS can be broadly classified into three categories based on the detection model employed: signature-based IDS (S-IDS), anomaly-based IDS (A-IDS), and hybrid-based IDS (H-IDS). S-IDS relies on predefined rules and signatures to detect attacks, while A-IDS utilizes machine learning techniques to identify legitimate behaviors within a system. H-IDS combines both A-IDS and S-IDS approaches [2, 20].

In this research, our focus is on anomaly-based IDS that employ machine learning techniques [22, 36, 37]. One widely used technique is clustering, which groups similar data points, also known as clusters [26]. Various clustering methods have been proposed for intrusion detection systems, including partitional, hierarchical, and other types of clustering [14, 23, 38]. Among these methods, k-means-based clustering is the most commonly applied due to its linear time complexity [25]. However, it suffers from drawbacks such as sensitivity to initial cluster centers and convergence to local optima [7]. To address the sensitivity of initial cluster centers, several optimization techniques have been incorporated into the clustering approach [31]. One such technique is Particle Swarm Optimization (PSO) [17], which has been successfully used to address this drawback in clustering tasks [9, 27].

However, clustering-based intrusion detection methods face challenges when dealing with scalability issues in the analysis of large volumes of network traffic. To address this, several parallel clustering methods have been developed in the literature to handle large-scale data [10]. Many of these methods leverage the MapReduce framework [12] for data processing. However, MapReduce is not well-suited for iterative algorithms as it requires frequent disk reads and writes, leading to performance limitations. To overcome the limitations of MapReduce, the Spark framework [44] has been proposed for efficient processing of iterative algorithms. Spark is an in-memory parallel framework that has the ability to process big data using a cluster of machines. In comparison to the MapReduce framework, Spark demonstrates greater efficiency and provides a significant speed improvement of approximately 10 to 100 times faster for data processing tasks [5]. This makes it a more suitable choice for handling large-scale data in intrusion detection systems.

In addition to addressing scalability, it is crucial to consider the need for interpretability in clustering-based intrusion detection methods. Security experts require an understanding of the rationale behind identifying certain operations as intrusions. Unfortunately, existing clustering methods for intrusion detection often lack interpretability making it challenging to explain the obtained clusters. These methods typically function as “black-box” systems, providing a final organization of network attacks without offering explanations for how the organization and attack detection were derived. However, explanations play an essential role in enabling security experts to transparently and effectively identify the correct attacks and implement appropriate strategies to secure the system. By incorporating interpretability into clustering-based intrusion detection methods, security experts can gain insights into the reasons behind classifying certain operations as security attacks and allow them to define effective security measures to protect the system.

To deal with all the discussed issues, we propose an Explainable Spark-based PSO Clustering for intrusion detection system (E-SPSO). The proposed system is based on the explainable artificial intelligence framework, named SHape Additive exPplanations (SHAP) [24], that allows explaining reasons behind the detected intrusions. Such explanations make the resulting prediction highly transparent for security experts. The proposed method performs parallel processing of intrusion detection tasks based on the Spark framework and integrates the PSO technique to improve the quality of obtained attack clusters. We will show in the next sections how the proposed intrusion detection system allows an enhance scalability, accuracy, and explainability of cyber-defense systems in large networks. The remainder of this paper is organized as follows: Section 2 discusses a survey of related works while Section 3 describes the backgrounds of this work including Particle Swarm Optimization (PSO), Spark framework, and Shapley Additive Explanations (SHAP). Then, Section 4 describes the proposed ESPSO-IDS system. Section 5 presents performed experiments and obtained empirical results. Finally, Section 6 summarises this work and discusses the future directions.

2. Related works

This section presents the related works of intrusion detection, explainable clustering, and explainable intrusion detection.

2.1. Intrusion detection

Several intrusion detection systems based on machine learning techniques were proposed in the literature [22, 37]. In this study, our focus is primarily on intrusion detection systems that utilize the clustering approach. Various clustering methods have been proposed specifically for intrusion detection systems [14, 19, 23, 38]. For instance, Li *et al.* [19] proposed a system that combines the k-means algorithm with PSO to create an effective intrusion detection system. This algorithm aims to benefit from the characteristics of PSO to overcome the premature convergence issue faced by the k-means algorithm. The proposed algorithm demonstrated relatively better results compared to the traditional k-means algorithm.

Indeed, Guan *et al.* [14] introduced a k-means-based clustering algorithm called y-means specifically for intrusion detection. y-means addresses the dependency of the k-means algorithm on the number of clusters. It aims to automatically partition a dataset into a reasonable number of clusters by classifying instances into "normal" and "abnormal" clusters. This approach provides a way to differentiate between normal and potentially intrusive activities. Similarly, Liu *et al.* [23] proposed an intrusion detection method based on the genetic clustering algorithm. This method consists of two stages: nearest-neighbor clustering and genetic optimization. The approach operates under the assumption that intrusion activities are likely to appear as outliers among normal activities and can be grouped into separate clusters from the normal cluster. This proposed method allowed to automatically establish clusters and detect intruders by labeling normal and abnormal groups respectively.

Although existing clustering-based intrusion detection methods demonstrate good performance, they often face limitations when applied to large-scale networks. To address this challenge, the parallelization of clustering algorithms has gained significant attention due to its effectiveness in reducing runtime in large networks. Parallel algorithms exploit multiple processing nodes to achieve a speedup compared to running the sequential version of the algorithm on a single processor. Several parallel clustering methods have been proposed to tackle scalability challenges in intrusion detection [4,15,41]. For example, Al-Jarah *et al.* [4] proposed IDS-MRCPSO, a parallel intrusion detection system based on the MapReduce framework. The proposed system integrates a PSO technique in the clustering step to improve the quality of intrusion detection. The use of PSO helps to overcome the sensitivity problem associated with initial cluster centers. Wu *et al.* [41] proposed a parallel intrusion detection system using the MapReduce framework. They combined the differential evolution algorithm with the k-medoids clustering algorithm to improve convergence efficiency in large networks. Additionally, they introduced a dynamic Gemini population schema to further enhance the optimization of the clustering step by maintaining solution diversity and avoiding local optima. Peng *et al.* [33] proposed an intrusion detection system based on the Mini Batch k-means clustering algorithm and Principal Component Analysis. Firstly, they employed a pre-processing method to digitize strings and normalize the data. Secondly, they applied Principal Component Analysis (PCA) [1] to reduce the dimensionality of the processed data. Finally, they incorporated the Mini Batch k-means [35] algorithm for data clustering. Recently, Ben HajKacem *et al.* [15] proposed a Spark-based intrusion detection system that integrates PSO for large-scale networks. This system offers a favorable trade-off between scalability and accuracy. The use of PSO clustering has allowed solving the sensitivity issue related to initial cluster centers as well as premature convergence.

Although many existing intrusion detection systems based on clustering techniques are effective in analyzing large amounts of data, they often fail to provide security experts with a way to understand and interpret the results they obtain. These methods often act as "black-boxes" that build clusters without offering any explanations for the underlying reasoning behind their formation. This lack of transparency makes it challenging to interpret the detected intrusions, particularly for non-domain experts, and significantly reduces user trust.

2.2. Explainable clustering

Explainable clustering, a sub-field of eXplainable Artificial Intelligence (XAI) [28], aims to address the issue of explainability by assisting decision-makers in interpreting the resulting clusters and providing insightful explanations. The demand for XAI has grown in recent years driven by the widespread use of machine learning models in critical domains that require explanations for their decision-making processes. Two main approaches to XAI methods have been proposed [3]: intrinsic XAI and post-hoc XAI. Intrinsic XAI approaches focus on explaining the structure and functioning of the model itself, but are limited to specific types of models. On the other hand, post-hoc approaches explain the final decisions of the model by analyzing the set of input data and can be applied to any model. Another classification of XAI techniques involves global (macro) and local (micro) explanations [8,21]. Global techniques aim to explain the general structure of the models by analyzing all of their decisions, whereas local techniques aim to provide explanations for individual decisions at the item level.

Explainable clustering methods follow a two-step process that utilizes XAI techniques to provide explanations for the clusters [6, 11, 29]. The first step focuses on assigning labels to the clusters, while the second step involves using these labels as target variables in a classification task. Explanations are then generated based on the resulting classification model. For instance, Morichetta *et al.* [29] proposed the EXPLAIN-IT method, which employs a supervised XAI technique to interpret clustering results. Initially, the authors cluster the input data using algorithms such as k-means or DBSCAN. Subsequently, a classifier is trained on the input data, employing the obtained cluster labels as the target variable. Finally, the classifier is explained using existing XAI models like LIME [34], which is commonly used to generate interpretations for individual predictions made by any classifier. Similarly, Horel *et al.* [16] also introduced a two-step method to explain the resulting clusters. First, a classifier is trained to assign cluster labels. Then, the Single Feature Introduction Test (SFIT) is applied to identify statistically significant features that characterize each cluster.

2.3. Explainable intrusion detection

Several works related to explainable intrusion detection systems were proposed in the literature [18, 39, 43]. Neupane *et al.* [32] conducted a study on existing explainable intrusion detection systems, which are primarily based on SHAP [24] and LIME [34] methodologies. Wang *et al.* [39] proposed a framework that utilizes SHAP to generate explanations for intrusion detection systems (IDS). This framework provides both local and global explanations to enhance the interpretation of IDS. The local explanation focuses on interpreting individual instances based on input features, whereas the global explanation reveals relationships between feature values and different attack types. Furthermore, Younisse *et al.* [43] also proposed an explainable intrusion detection system by combining deep neural networks with interpretable model predictions. Their system utilizes SHAP to provide both local and global explainability for improving the interpretation of IDS. In addition, in the context of Internet of

Things (IoT) networks, Keshk *et al.* [18] proposed an explainable intrusion detection system. They developed an IDS using a Long Short-Term Memory (LSTM) model for cyber-attack identification and utilized the SPIP (S: Shapley Additive exPlanations, P: Permutation Feature Importance, I: Individual Conditional Expectation, P: Partial Dependence Plot) technique to produce explanations for the model's decisions. The proposed system achieved high detection accuracy, efficient processing time, and improved interpretability compared to other IDS systems.

Despite the significant efforts to enhance the transparency and explainability of IDS, there are still several limitations and open challenges that need to be addressed. One major challenge is generating explanations in large-scale IDS systems. The vast amount of data in such systems poses difficulties in maintaining scalability while ensuring IDS accuracy. Developing techniques that can handle the volume and velocity of data in real-time while still providing meaningful and interpretable explanations, remains an interesting and complex challenge. Additionally, most of the existing explainable intrusion detection systems are built for IDS based on supervised classification approaches. These systems require labeled data for training and rely on predefined attack types. However, in real-world scenarios, there may be unknown or novel attack types that have not been labeled or encountered before. Therefore, exploring the explainability of IDS based on unsupervised approaches is an important area that has yet to be extensively studied in the literature.

3. Background

3.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) was originally proposed by Kennedy and Eberhart in 1995 [17]. PSO is inspired by the behavior of birds in a flock. It mimics their social interactions while searching for food. The algorithm has been widely applied to solve various optimization problems. In PSO, a population of particles forms a swarm. Each particle represents a potential solution to the optimization problem. At time t, each particle P_i is characterized by its current position $x_i(t)$ in the search space, its velocity $v_i(t)$, and its personal best position $pbestP_i(t)$ along with the corresponding fitness value $pbestF_i(t)$. The personal best position of a particle represents the best solution it has encountered and is defined as follows:

$$pbestP_i(t+1) = \begin{cases} pbestP_i(t)\ if\ f(pbestP_i(t)) <= f(x_i(t+1)) \\ \\ x_i(t+1)\ if\ f(pbestP_i(t)) > f(x_i(t+1)) \end{cases} \tag{1}$$

The global best position represents the best fitness value of any particle and is defined as follows:

$$gbestP(t+1) = min\ (f(y), f(gbestP(t))) \tag{2}$$

where $y \in \{pbestP_0(t), ..., pbestP_S(t)\}$.

The particle position and velocity are updated using the following formula:

$$x_i(t+1) \leftarrow x_i(t) + v_i(t) \tag{3}$$

$$v_i(t+1) \leftarrow wv_i(t) + c_1r_1(pbestP_i(t) - x_i(t)) + c_2r_2(gbestP(t) - x_i(t)) \tag{4}$$

where w is the inertia weight, $x_i(t)$ is the position of the particle P_i at time t, $v_i(t)$ is the velocity of the particle P_i at time t, c_1 and c_2 are two acceleration coefficients, and r_1 and r_2 are two random values in the range $[0, 1]$. The main algorithm of PSO is shown in Algorithm 1. The algorithm begins by creating an initial population of particles from the input dataset R. Then, it enters a loop until the convergence criteria are met. Within each iteration, the fitness value of each particle is calculated. The personal best position of each particle is updated using Equation (1). The global best position is updated using Equation (2). The velocities and positions of the particles are then updated using Equations (3) and (4), respectively. The algorithm continues iterating until the convergence criteria are reached.

Algorithm 1 PSO main algorithm

```
Input: Input dataset $R$
Output: Particle information
Create an initial population of particles from $R$.
while Convergence not reached do
    Calculate the fitness value of particles.
    Update the personal best position of each particle using Equation (1).
    Update the global best position using Equation (2).
    Update the velocities and positions using Equations (3) and (4), respectively.
end while
```

3.2. Spark framework

MapReduce [12] is a parallel programming framework based on the *map* and *reduce* phases. Each phase involves input and output $< key/value >$ pairs. During the map phase, map functions are executed in parallel to process each *key* and *value* pair and lead to the generation of a collection of intermediate $< key'/value' >$ pairs. Then, the shuffle phase compiles a list of all intermediate values associated with a particular intermediate key. After that in the reduce phase, the reduce function merges all intermediate values that belong to the same intermediate key. The data flow of the MapReduce framework is depicted in Figure 1. Input and output data for MapReduce are stored in a distributed file system accessible from the cluster of machines. Hadoop has implemented the MapReduce framework [40] and provides a distributed file system called Hadoop Distributed File System (HDFS) for data storage on the machines. Despite its high performance, the MapReduce framework is not suitable for iterative algorithms. The need to read and write data from disks in each iteration can significantly decrease the algorithm's efficiency.

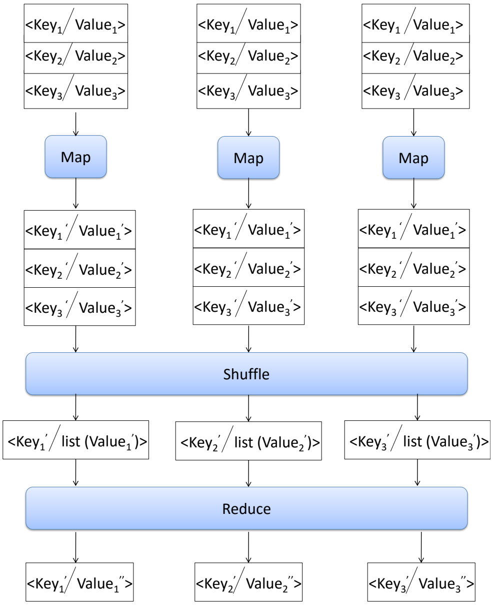


Figure 1. Data flow of MapReduce framework

Spark [44] is a parallel framework and one of the most widely used Big Data parallel processing frameworks. The high efficiency of Spark is due to resilient distributed datasets (RDDs) allowing to perform in-memory processing. RDDs can be stored in memory and used in multiple consecutive operations. Spark is integrated with Hadoop allowing it to read data from Hadoop Distributed File System (HDFS). Moreover, Spark provides a set of in-memory operators that enable faster data processing on distributed environments compared to standard MapReduce. The Spark framework supports two types of operators that can be applied to RDDs: transformations and actions. Transformations are used to apply a function to all elements of an RDD and return new RDDs. Actions, on the other hand, either return a value to the program or write the computation result to an external storage. This work employed the following Spark functions:

- `mapToPair(func)`: Creates a new RDD by applying the function `func` to all elements of the RDD.
- `flatMapToPair(func)`: Creates a new RDD by applying the function `func` to each element of the RDD and merging the results.
- `groupByKey(nums)`: Groups and distributes the values for each key in the RDD into a single sequence.
- `partitionBy(Partitioner)`: Creates a copy of the RDD partitioned using the specified partitioner.

- `filter(condition)`: Creates a new RDD that stores only the elements satisfying a given condition.
- `collect()`: Creates an array that stores all the elements in a particular RDD.
- `saveAsTextFile("...")`: Saves the elements of the RDD in a text file at the specified file path directory.

By utilizing these functions, Spark enables efficient and flexible data processing in distributed environments.

3.3. SHAP (SHapley Additive exPLanations)

SHAP is a XAI technique which is used to analyze predictions made by machine learning models [24]. It is based on game theory and provides explanations by detecting how each feature contributes to the accuracy of the predictions. SHAP also provides the most important features and their impact on model prediction. It deals with the Shapley values to evaluate each feature's impact on the machine learning prediction model. Shapley value is calculated as the (weighted) average of marginal contributions. It is defined by the impact of feature value on the prediction overall potential feature coalitions. Shapley value for an instance x is computed as follows:

$$\phi_{j_r}(x) = \sum_{S \subseteq \{j_1 \cdots j_m\} \setminus \{j_r\}} \frac{\mid S \mid ! \times (m- \mid S \mid -1)!}{m!} \times \delta_{j_r} \tag{5}$$

with

$$\delta_{j_r} = [f_{S \cup j_r}(x) - f_S(x)] \tag{6}$$

where $\phi_{j_r}(x)$ represent the Shapley value for feature value with the index $j_r \in [1, ..., m]$, S is a subset of the features employed in the prediction model, $|S|$ is the carnality of S, m is the number of features, $f_{S \cup j_r}(x)$ and $f_S(x)$ are the prediction function for the set of feature values in S with and without including the feature j_r respectively. The Shapley value $\phi_{j_r}(x)$ quantifies how much the feature j_r influences the prediction model, either positively or negatively. To this end, the model is trained with and without including this feature and then predictions from the two models are compared for all subset features $S \subset \{j_1, ..., j_m\} \setminus \{j_r\}$. A large positive value for $\phi_{j_r}(x)$ indicates that the feature j_r has a significant positive influence on the prediction. However, a large negative value for $\phi_{j_r}(x)$ shows that the feature j_r has a significant negative contribution on the prediction.

4. Proposed explainable Spark-based PSO clustering method for intrusion detection

In order to simultaneously solve the issues of large amounts of network traffic data and the complexity of explaining the built intrusion detection model, we propose a new design of an explainable Spark-based PSO clustering approach for intrusion detection.

As shown in Figure 2, E-SPSO consists of three main phases: *data detector modeling phase*, *data labeling phase* and *model explaining phase*.

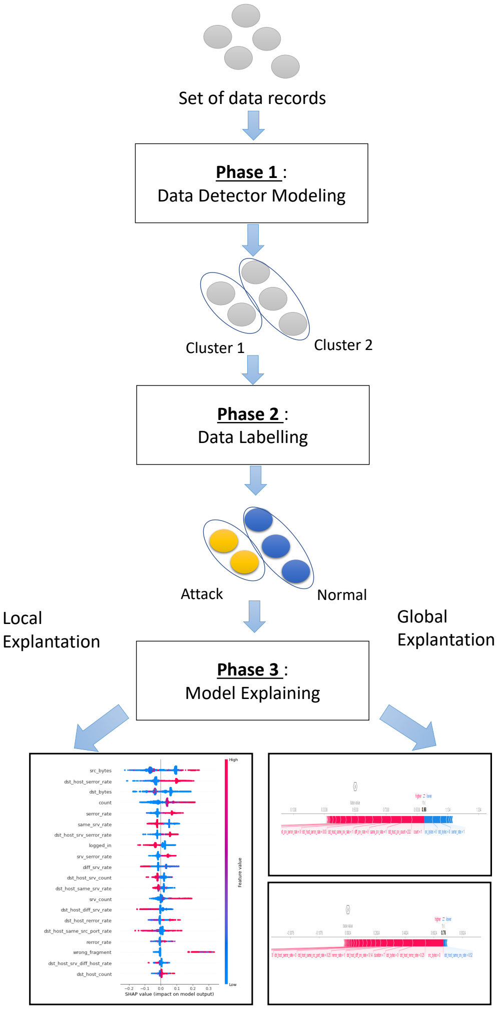


Figure 2. The main phases of the proposed explainable Spark-based PSO clustering approach for intrusion detection. Phase 1: Data detector modeling, phase 2: Data labeling and phase 3: Model explaining

In the first phase, we propose to exploit a Spark-based PSO clustering method [30] to guide the intrusion detection model by generating global best centers of obtained clusters. The use of particle swarm optimization for the clustering task is a very efficient way since particle swarm optimization avoids the sensitivity problem of initial cluster centers as well as premature convergence. In the second phase, we assign a cluster label to testing data by computing distances between the testing data and the final global best centers. In the third phase, we propose to explain the built intrusion detection model by exploiting XAI SHAP method capabilities. In fact, the transparency and the easy interpretation of the intrusion detection model are almost as important as the classification accuracy. Hence, an intelligent SHAP-based process is designed having as outputs detailed explanations of the global structure of the intrusion detection model as well as local explanations regarding the assignment of each connection to any specific class.

4.1. Phase 1: data detector modeling

The data detector modeling phase consists of applying the parallel particle swarm optimization clustering method through the Spark framework. The use of particle swarm optimization in the clustering task is an efficient solution to avoid the sensitivity problem of initial cluster centers. The proposed method is a partitioning clustering type that uses representatives, called also centers, to model the clusters. The initial centers are selected randomly from the input data, then the centers are updated based on the swarm particle velocities until the convergence is achieved. The best cluster centers are then used in the data labeling phase by computing the average minimum distances between the data and the cluster centers.

It is important to note that the proposed method stores for each particle the following information: *position vector*, *velocity vector* and *fitness value*. The particle information is updated in each iteration using the information from the previous iteration. The proposed method is composed of three MapReduce jobs namely, *Data assignment and fitness computation*, *Personal and global best update* and *Position and velocity update*.

4.1.1. Data assignment and fitness computation

In the first MapReduce job, E-SPSO starts by initializing particle information. The positions of particles are randomly selected from the input data as initial cluster centers. Then, the data is divided into chunks where each chunk is assigned to a map function. The particle information are then transferred to all chunks. The map function first assigns each data record to the nearest cluster center in each particle by computing distances. Then, the map function generates a key-value pair as output where the key represents the couple $< particleID, centerID >$ and the value represents the minimum distance between a single data and the centerID in a particleID. Once all data are assigned to the nearest cluster, a reduce function is applied to compute the fitness value by merging data from different map functions. The fitness value is computed as follows:

$$Fitness = \frac{\sum_{j=1}^{k} \sum_{i=1}^{|C_j|} dis(d_i, C_j)}{k} \tag{7}$$

with $dis(d_i, C_j)$ represents the distance between a data record d_i and the cluster center C_j. $|C_j|$ represents the number of data records assigned to the center C_j, and

k represents the total number of clusters. The reduce function in the MapReduce job generates key-value pairs as output where the key represents the particle ID, and the value represents the fitness value.

Let $D = \{d_1, d_2, ..., d_n\}$ be the set of data records. Let $P(t) = \{P_1(t), P_2(t), ..., P_S(t)\}$ be the set of particle information, where $P_c(t) = \{x_c(t), v_c(t), pbestP_c(t), pbestF_c(t)\}$ represents the information of particle c in iteration t. Here, $x_c(t)$ is the position, $v_c(t)$ is the velocity, $pbestP_c(t)$ is the best position, and $pbestF_c(t)$ is the best fitness of particle c. Let $F = \{F_1, F_2, ..., F_S\}$ be the set of fitness values, where F_c represents the fitness value of particle c. The main steps of Data assignment and fitness computation MapReduce job are described in Algorithm 2.

Algorithm 2 Data assignment and fitness computation MapReduce job

```
Input: Input dataset D
Output: Fitness values F
$P(t)$ ← Initialize particle information from $D$
Divide the data $D$ into $m$ RDD $D = \{D^1 ... D^m\}$
% Map Phase
Let $D^p$ be an RDD assigned to map task $p$.
for each $d_i \in D^p$ do
    for each $P_c \in P(t)$ do
        $x_c(t)$ ← Extract positions from $P_c(t)$
        Assign each data point to its nearest cluster centroid by computing distances.
        Let mindis the minimum computed distance.
        Let CentroidID the index of the cluster centroid where the data point $r_i$ is assigned.
        Let ParticleID the index of the particle $P_c$.
    end for
    Emit (key: ParticleID, CenterID/value: mindis)
end for
% Reduce Phase
for each $P_i \in P(t)$ do
    Calculate fitness value $F_i$ using Equation (7).
    Emit (key: ParticleID /value: $F_i$)
end for
```

4.1.2. Updating pbest and gbest update

Once all particle fitness values are computed, they are automatically distributed to RDD collections. Given that the computation of pbest (personal best) and gbest (global best) is not an expensive operation, they are computed locally without using the parallel framework. Let $pbestF(t) = \{pbestF_1(t), pbestF_2(t), ..., pbestF_S(t)\}$ be the set of personal best fitness values where $pbestF_i(t)$ represents the pbest fitness of particle i at iteration t. Let $pbestP(t) = \{pbestP_1(t), pbestP_2(t), ..., pbestP_S(t)\}$ be the set of personal best positions where $pbestP_i(t)$ represents the pbest position of particle i at iteration t. Let $gbestP$ be the position of the best particle. The main steps of the pbest and gbest update MapReduce job are described in Algorithm 3.

Algorithm 3 pbest and gbest update MapReduce job

```
**Input:** F, $pbestF(t)$, $pbestP(t)$
**Output:** $pbestF(t+1)$, $pbestP(t+1)$, $gbestP$
$gbestP \leftarrow \emptyset$
**for** each $P_i(t) \in P(t)$ **do**
    $pbestF_i(t+1) \leftarrow \emptyset$
    $pbestP_i(t+1) \leftarrow \emptyset$
    **if** $(pbestF_i(t) \leq F_i)$ **then**
        $pbestF_i(t+1) \leftarrow pbestF_i(t)$
        $pbestP_i(t+1) \leftarrow pbestP_i(t)$
    **else**
        $pbestF_i(t+1) \leftarrow F_i$
        $pbestP_i(t+1) \leftarrow x_i(t+1)$
    **end if**
**end for**
Let $i^*$ is the index of a particle having the best fitness value.
$gbestP \leftarrow x_{i^*}(t)$
```

4.1.3. Position and velocity update

During the MapReduce job, the E-SPSO algorithm begins by assigning particle information to different map functions. Each map function then performs velocity and position updates using Equations (3) and (4). The reduce function groups all the intermediate key-value pairs computed by the map functions. After the reduce phase is completed, the particle information is distributed among RDD collections which are stored in memory for the next iteration. Let $x(t) = \{x_1(t), x_2(t), ..., x_S(t)\}$ represent the set of position values, where $x_i(t)$ denotes the position of particle i at iteration t. Similarly, let $v(t) = \{v_1(t), v_2(t), ..., v_S(t)\}$ denote the set of velocity values where $v_i(t)$ represents the velocity of particle i at iteration t. The main steps of the Position and Velocity Update MapReduce job are described in Algorithm 4.

Algorithm 4 Position and Velocity update MapReduce job

```
**Input:** $gbestP$, $P(t)$
**Output:** $P(t+1)$
% *Map Phase*
Divide the data $P(t)$ into $m$ RDD $D = \{P^1...P^m\}$ Let $P^p(t)$ be an RDD assigned to a
    map task $p$.
$x_i(t+1) \leftarrow \emptyset$
$v_i(t+1) \leftarrow \emptyset$
Update the new position value $x_i(t+1)$ using Equation 4
Update the new velocity value $v_i(t+1)$ using Equation 3
Emit(key: 1/value: $P_i(t+1)$)
% *Reduce Phase*
Group outputs from the different map functions and update the new particle information
    $P(t+1)$
Emit $(P(t+1))$
```

4.2. Phase 2: data labeling

Once the data detector modeling phase is completed, the global best cluster centers are extracted from the final particle's information. In this phase, the detection model is evaluated by computing distances between the testing data and the global best centers. To accomplish this, the testing data is assigned to their nearest clusters based on the computed distances.

The next step involves the cluster labeling process where the correct labels are predicted for the clusters generated during the testing data assignment. Cluster labeling is performed by determining the maximum intersection percentage between the true labels of the testing data and the assigned clusters generated during the testing data assignment phase. The main steps of the labeling phase are described in Algorithm 5.

Algorithm 5 Data labelling phase

```
**Input:** Testing data $T$, Final Particle information $P$
**Output:** Labelled data
Let $C()$ the $k$ centers extracted from the final particle $P$.
**for** each $t_i \in T$ **do**
    Compute distances between $t_i$ and $C$.
    Assign $t_i$ to its nearest center.
**end for**
Apply the cluster labeling.
```

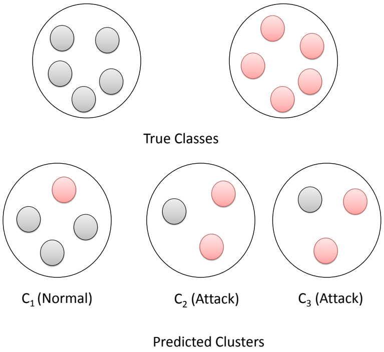


Figure 3. An illustrative example of the clusters labeling process

Figure 3 provides an illustrative example to aid in understanding the cluster labeling process. Let's examine the labeling process for each cluster. Cluster C_1 consists of $3/4$ normal connections and $1/4$ attack connections. As a result, it is labeled as "normal connections". Cluster C_2 comprises $1/3$ normal records and $2/3$ attack records, leading to its labeling as "attack cluster". Similarly, cluster C_3 contains $1/3$ normal connections and $2/3$ attack connections, resulting in its classification as an "attack cluster".

4.3. Phase 3: model explaining

This phase aims to build detailed explanations on the built classes resulting from the previous phase. These explanations allow cybersecurity experts to better understand and interpret the resulting classification of the connections in terms of local and global levels. Local explanations have the objective to explain the reasons behind the prediction of each connection to any class in terms of feature values while global explanations try to explain important feature values at the level of classes. This phase is based on explaining predictions using the SHAP method capabilities. First, the resulting classes are used as the label class variables of the explainable process. Then, local and global explanations are generated by applying the SHAP technique.

Concerning local explanations, we calculate for each record and each feature the Shapley value regarding the assigned class. These Shapley values measure how much each feature, for each record, contributes to the final prediction. These local explanations support cybersecurity experts to better understand the reasons behind assigning a connection to any specific normal or attack class. Experts can also analyze the positive and negative contribution of each feature of each connection on the predicted class label. Concerning global explanations, we calculate further explanations regarding the importance and contributions of the features when building each class. We use local Shapley values of each data record as a "single unit" to build an overall score for each class. The global explanation is performed as follows:

$$G_j = \frac{1}{n} \sum_{i=1}^{n} |\phi_j(x_i)| \tag{8}$$

where $\phi_j(x_i)$ is the Shapley value of the feature j for the record x_i, G_j refers to the overall Shapley value of the feature j and n is the total number of records in the dataset. Global shapley values are then sorted in decreasing order to generate the most important feature of the model.

5. Experiments and results

5.1. Dataset description

In order to evaluate the performance of the proposed method, we used a Big intrusion detection dataset[1] which is well suited for intrusion detection problems. This dataset contains a standard set of data records which includes a wide variety of normal and attack connections in a military network environment. Each record in the dataset represents a connection between two IP addresses. The data is classified into normal traffic and four kinds of attacks namely, denial of service (DOS), probe (PROB), remote to local (R2L) and user to root (U2R). Each connection in the dataset is described by three categorical and 38 numerical features for a total of 41 features. The

[1] https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

detailed descriptions of the features are given in [13]. The training dataset contains $4,898,431$ data records collected during seven weeks of network traffic while the testing dataset contains $311,029$ records collected during two weeks.

In order to evaluate the scalability of the proposed method, we extract 4 different data samples from the whole training data set. To simplify the names of the data samples, we will use the notations Train20, Train40, Train80, and Train100 to denote an extracted data set that stores 20%, 40%, 80%, and 100% of the whole training data set. Statistics of these datasets are summarized in Table 1.

Table 1

Summary of the data samples

Dataset	Number of records	Normal	Attack
Train20	979,686	194,556	785,130
Train40	1,959,372	389,112	1,570,260
Train80	3,918,745	778,225	3,140,520
Train100	4,898,431	972,781	3,925,650

5.2. Data pre-processing

A set of pre-processing techniques was applied to the training and testing datasets. Firstly, records with missing values were eliminated as distances cannot be computed for these records. Additionally, columns containing categorical features were removed to retain only the numerical features. Subsequently, a min-max normalization within the range of $[0, 1]$ is applied to the obtained dataset. This normalization step helps address bias issues arising from features that exhibit significant variation between their minimum and maximum values. The normalization is performed using the following equation:

$$x_{ij_{new}} = \frac{x_{ij} - x_{j_{min}}}{x_{j_{max}} - x_{j_{min}}} \tag{9}$$

where x_{ij} is the value of record i for feature j, $x_{ij_{new}}$ is the normalized value of record i for feature j, $x_{j_{min}}$ is the minimum value of feature j and $x_{j_{max}}$ is the maximum value of feature j.

5.3. Evaluation measures

In order to evaluate the scalability of the proposed method, we use the Speedup measure [42] which consists of fixing the dataset size and varying the number of machines. The Speedup measure is defined as follows:

$$Speedup = \frac{T_1}{T_m} \tag{10}$$

where T_1 is the running time of processing data on 1 machine and T_m is the running time of processing data on m machines.

In order to evaluate the quality of the proposed method, we use true positives, true negatives, false positives, and false negatives. A true positive (TP) indicates that the intrusion detection system detects precisely a particular attack that has occurred. A true negative (TN) indicates that the intrusion detection system has not made a mistake in detecting a normal connection. A false positive (FP) indicates that a particular attack has been detected by the intrusion detection system but that attack did not actually occur. A false negative (FN) indicates that the intrusion detection system is unable to detect the intrusion after a particular attack has occurred.

We use in this paper the True Positive Rate (TPR) and False Positive Rate (FPR) that are defined in Equation (11) and Equation (12) respectively.

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

Furthermore, we use the Area Under Curve (AUC) measure [46] to combine the TPR and FPR which is considered a good indicator of these rates. The AUC can be defined as follows:

$$AUC = \frac{(1 - FPR) \cdot (1 + TPR)}{2} + \frac{FPR \cdot TPR}{2} \tag{13}$$

A greater value of these measures indicates better quality results.

5.4. Evaluation of the clustering quality

We evaluate the accuracy of the proposed method (E-SPSO) compared to four existing methods: k-means [25], PSO [27], MRKM [45] and MRPSO [4]. We use in the experiments the following parameters: the number of particles equal to 10, the number of iterations equal to 50, the inertia weight equal to 0.72 and the acceleration coefficients equal to 1.49. Table 2 reports the TPR, FPR and AUC values obtained by the proposed method using different training data sample sizes. The obtained results show that the proposed E-SPSO gives nearly the same results as the existing MRPSO. In addition, we observed that TPR value of E-SPSO using the whole training data (i.e Train100) reaches its best value compared to those obtained in smaller training datasets. Furthermore, Table 2 shows that E-SPSO obtains the lowest FPR for Train100 dataset.

Table 2
Comparison of the accuracy of E-SPSO with existing methods

Dataset	Method	TPR	FPR	AUC
Train20	k-means	0.789	0.214	0.758
	PSO	0.879	0.015	0.927
	MRKM	0.789	0.214	0.758
	MRPSO	0.903	0.038	0.933
	E-SPSO	0.848	0.096	0.875
Train40	k-means	0.876	0.155	0.733
	PSO	0.841	0.019	0.901
	MRKM	0.876	0.155	0.733
	MRPSO	0.911	0.021	0.945
	E-SPSO	0.856	0.085	0.902
Train80	k-means	0.798	0.087	0.828
	PSO	0.899	0.113	0.914
	MRKM	0.798	0.087	0.828
	MRPSO	0.935	0.013	0.961
	E-SPSO	0.879	0.068	0.944
Train100	k-means	0.871	0.149	0.719
	PSO	0.939	0.013	0.963
	MRKM	0.871	0.149	0.719
	MRPSO	0.939	0.013	0.963
	E-SPSO	0.883	0.059	0.905

For instance, the E-SPSO has a high TPR value (0.883) for Train100 while it is equal to 0.848 for Train20 dataset. In addition, E-SPSO has a low FPR of 0.059 for Train100 compared to the value of 0.096 for Train20 dataset. The obtained high scores of TPR and the low values of FPR show that the proposed method can effectively distinguish between normal and attack data records. Better scores are shown as the size of the training dataset increases.

5.5. Evaluation of the scalability

We firstly evaluated the running time of the proposed method compared to the existing methods. Figure 4 shows the obtained running times for the 4 training data samples. The obtained results show that the proposed method is faster than existing methods. For instance, the E-SPSO is faster by a factor of 3.72 and 1.04 than PSO and MRPSO respectively for Train100 dataset. This result can be explained by the effectiveness of the parallel framework Spark when processing large-scale data. Hence, we can conclude that the parallel PSO clustering method based on Spark framework can be a good solution to handle large intrusion detection problems.

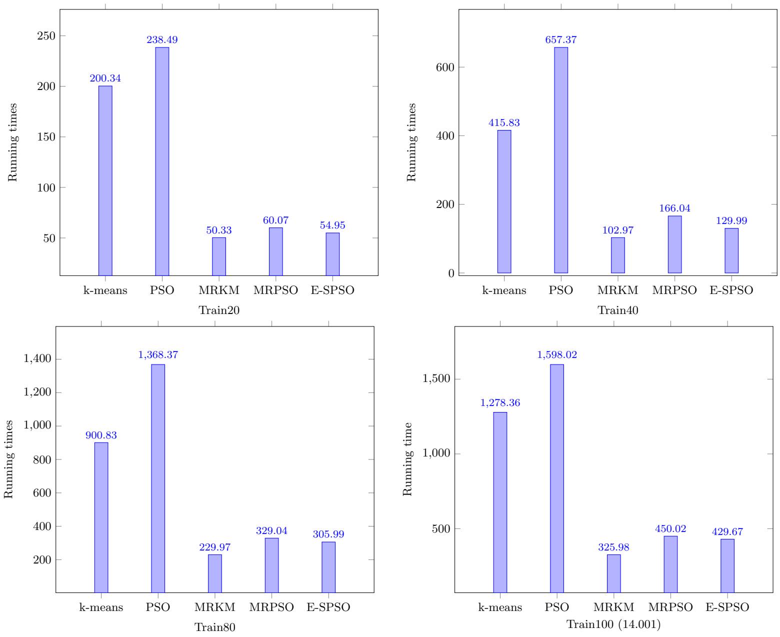


Figure 4. Comparison of the running times of E-SPSO with existing methods.

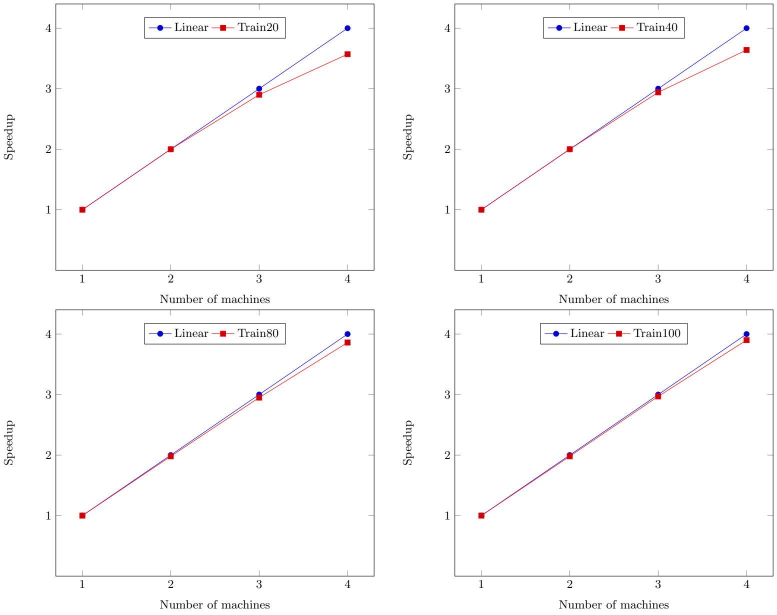


Figure 5. Evaluation of Speedup results using an increasing number of machines

After that, we evaluated the scalability of the proposed method using an increasing number of machines. Figure 5 shows the Speedup results using different training data sizes with different numbers of machines. This figure shows good Speedup results as the data size increases. For example, the Speedup value when running E-SPSO using 4 machines for Train20 is 3.57 while it is 3.90 for Train100 data. In addition, the proposed method shows approximately a linear speedup when the number of machines increases. This can be explained by the benefits of the in-memory processing of the Spark framework which can significantly reduce the network cost when the number of machines increases.

5.6. Evaluation of the local explanation

To provide local explanations, we utilized a force plot diagram to visually illustrate the contribution of each feature in the decision-making process of the model. The length of each feature's representation along the horizontal axis indicates its importance in the decision-making process. The colors red and blue are used to indicate the influence of a particular feature on the decision to assign records to a specific class. The red color signifies that a feature with a specific value may boost the assignment to that class. On the other hand, the blue color indicates that a feature with a specific value boosts the assignment to the other classes.

In Figure 6, instead of analyzing a single record for local feature importance, we chose to randomly select 10 data records from each class. This approach provides a more significant analysis of the feature contributions as it considers a broader range of records for each class.

Figure 6a presents the force plot diagram for an average of 10 data records that were correctly classified by the model as a DOS attack. This figure shows that the feature *same_srv_rate* plays a significant role in determining the assignment of records to the DOS attack class, with a value of 0.01. This lower value of *same_srv_rate* increases the probability of assigning the record to the DOS attack class. Furthermore, the features *diff_srv_rate* with a value of 0.08 and *count* with a value of 113 also result in a higher probability of assigning the data record to the DOS attack class.

Besides, Figure 6b focuses on an average of 10 data records classified as PROB attacks. The figure highlights that features such as *src_bytes = 0*, *dst_host_rerror_rate = 0.25*, and *dst_bytes = 0* contribute significantly to the decision of classifying a record as a PROB attack. Notably, the feature *dst_host_same_srv = 0.52* reduces the probability of assigning a record to the PROB attack class. Moving on to Figure 6c that provides a local explanation for 10 data records that were correctly classified as R2L attacks, the features *num_compromised=2*, *hot = 3*, and *src_bytes = 2.42* play a significant role in the decision to classify a record as an R2L attack. Higher values of these features greatly increase the likelihood of the record being classified as an R2L attack. Concerning Figure 6d that presents a local explanation for 10 data records that were correctly assigned to the U2R attack class, the feature *duration* with a value of 12 contributes positively to the final classification, indicating that a longer duration increases the likelihood of the connection being assigned to the U2R attack class. Additionally, features such as *dst_bytes* = *0* and *src_bytes* = *7.045* also positively contribute to this decision.

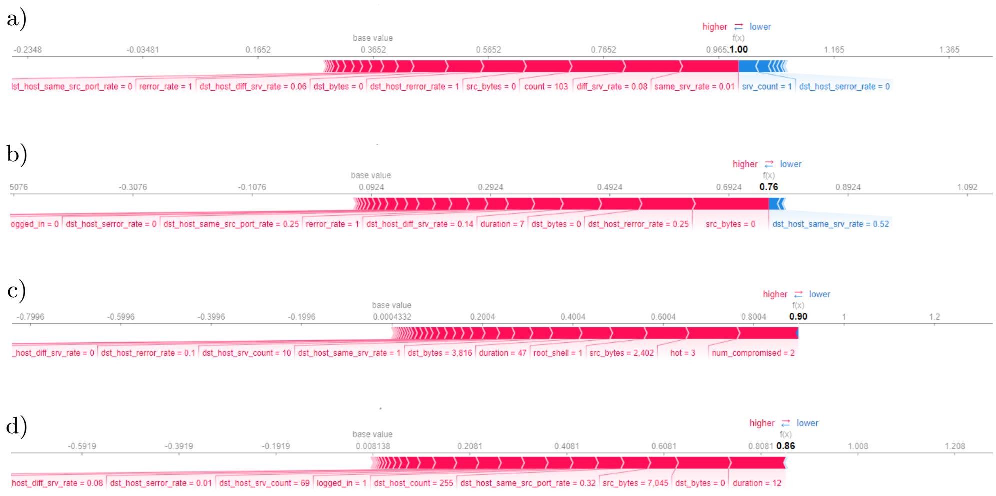


Figure 6. Force plot diagram illustrating local feature contribution in the decision of assigning records to each attack class, based on an average of 10 data records per class: (a) class DOS, (b) class PROB, (c) class R2L, (d) class U2R. The varying colors, red and blue, indicate whether a feature increases or decreases the probability of assigning records to a particular attack class

5.7. Evaluation of the global explanation

The global explanation of E-SPSO results aims to provide detailed insights into the key features that played a crucial role in the classification of attack classes. To build such explanations, we used the SHAP feature-summary-plot as shown in Figure 7. Each dot in the figure represents a data record from the used dataset. The dot's vertical position represents the feature while the horizontal position represents the impact of that feature value on the model's classes (local to each class). The color of each dot indicates the value of the corresponding feature for that record, with red representing high values, purple representing medium values, and blue representing low values.

Figure 7a illustrates that high values of the *src_bytes* feature increase the probability of the connection being classified as a DOS attack by 20% to 30%. Conversely, low values of the *dst_bytes* feature increase the probability of the model identifying the records as DOS attacks. Figure 7b demonstrates the global explanation of PROB attacks. Low values of the *src_bytes* feature increase the probability of a prediction as a PROB attack by 1% to 30%, while large values of the *dst_host_same_srv* feature increase the probability of the connection being classified as a PROB attack by 10% to 30%.

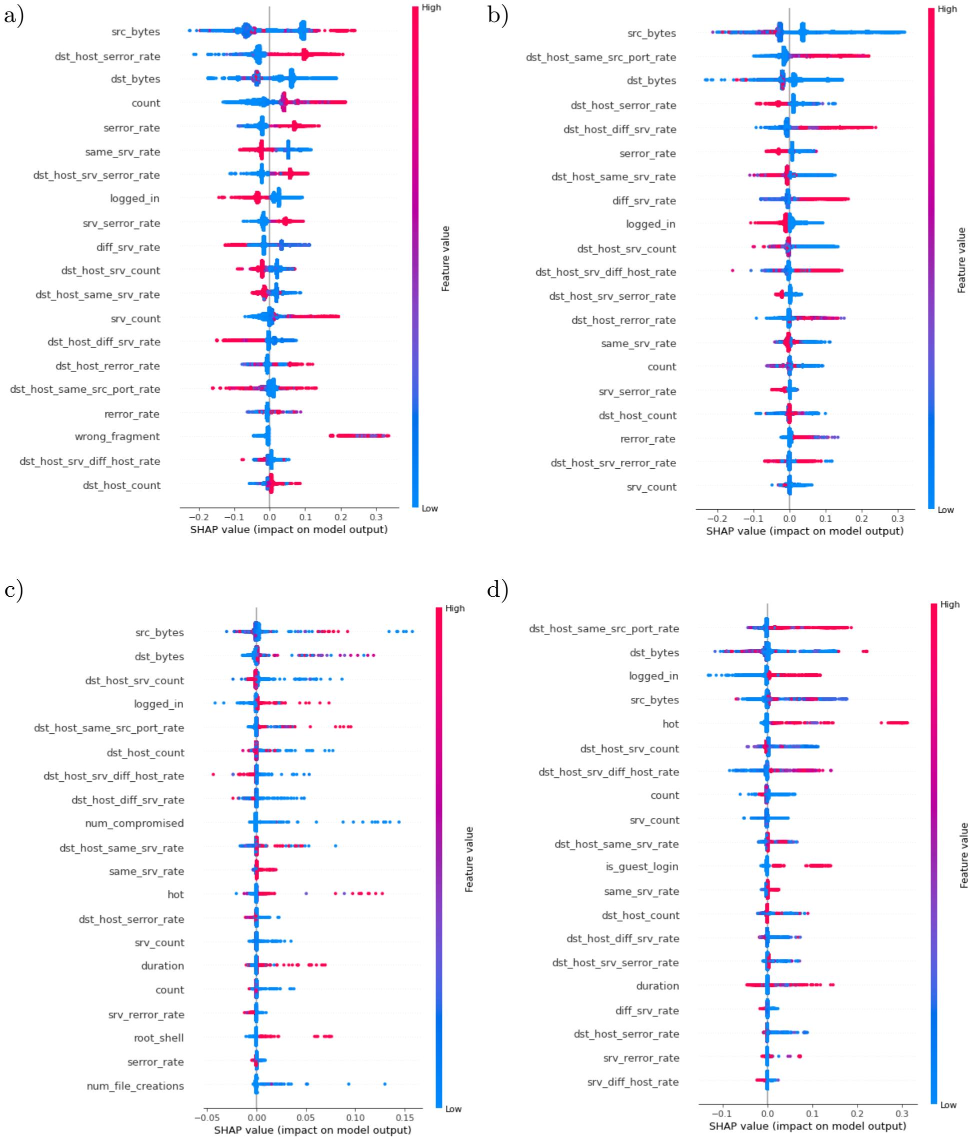


Figure 7. Top 20 important features that were crucial in building each attack class: a) class DOS; b) class PROB; c) class R2L; d) class U2R

Then, Figure 7c provides the global explanation of R2L attacks. Low values of the *dst_srv_count* feature increase the probability of a prediction as an R2L attack by 1% to 30%. Additionally, the figure shows that large values of the *logged_in* feature can increase the probability of the connection being classified as an R2L attack by up to 30%. The last sub-figure, Figure 7d, illustrates the global explanation of U2R attacks. Low values of the *dst_srv_count* feature can increase the probability of the records being considered as U2R attacks by up to 30%. Moreover, large values of the *dst_host_same_src* and *logged_in* features can increase the probability of the connection being classified as a U2R attack by up to 20%.

6. Conclusion

We proposed in this work an explainable Spark-based PSO clustering for an effective intrusion detection system that deals with the issues of scalability, accuracy, and explainability of intrusion detection. The proposed system uses parallel clustering and explainable artificial intelligence capabilities to build local micro and macro explanations. It includes three independent phases: data detector modeling, data labeling, and model explaining. The first phase aims to build clusters based on Spark-based PSO clustering, the second phase assigns an attack label to each build group while the third phase is devoted to the generation of local and global explanations. Empirical experiments performed on real-world data from the military environment have shown a significant improvement of the scalability, accuracy, and explainability of intrusion detection.

The output of the proposed system relies on the quality of defined features. The initial selection of features on such applications has a large impact on model outputs and also on the explainability of results. It would be interesting to include a pre-step of feature selection to extract the most important features when building attack clusters. Besides, we considered in this work a pre-configured number of clusters (known in advance). However, in real-life intrusion detection problems, it would be interesting to integrate automatic techniques for estimating the number of existing attack classes. Another interesting future direction to improve this work is to integrate other XAI techniques such as LIME which may improve the model interpretability.

Acknowledgements

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-23-DR-101). Therefore, the authors thank the University of Jeddah for its technical and financial support

References

[1] Abdi H., Williams L.J.: Principal component analysis, *WIREs Computational Statistics*, vol. 2(4), pp. 433–459, 2010. doi: 10.1002/wics.101.

[2] Ahmad Z., Shahid Khan A., Wai Shiang C., Abdullah J., Ahmad F.: Network intrusion detection system: A systematic study of machine learning and deep learning approaches, *Transactions on Emerging Telecommunications Technologies*, vol. 32(1), e4150, 2021. doi: 10.1002/ett.4150.

[3] Ali S., Abuhmed T., El-Sappagh S., Muhammad K., Alonso-Moral J.M., Confalonieri R., Guidotti R., *et al.*: Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence, *Information Fusion*, 101805, 2023. doi: 10.1016/j.inffus.2023.101805.

[4] Aljarah I., Ludwig S.A.: Towards a scalable intrusion detection system based on parallel PSO clustering using mapreduce. In: *GECCO '13 Companion: Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, pp. 169–170, 2013. doi: 10.1145/2464576.2464661.

[5] Awan M.J., Rahim M.S.M., Nobanee H., Yasin A., Khalaf O.I., Ishfaq U.: A Big Data Approach to Black Friday Sales, *Intelligent Automation & Soft Computing*, vol. 27(3), pp. 785–797, 2021. doi: 10.32604/iasc.2021.014216.

[6] Bandyapadhyay S., Fomin F., Golovach P.A., Lochet W., Purohit N., Simonov K.: How to Find a Good Explanation for Clustering? In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36(4), pp. 3904–3912, 2022. doi: 10.1609/aaai.v36i4.20306.

[7] Bradley P.S., Fayyad U.M.: Refining initial points for k-means clustering. In: *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, vol. 98, pp. 91–99, Citeseer, 1998.

[8] Carvalho D.V., Pereira E.M., Cardoso J.S.: Machine learning interpretability: A survey on methods and metrics, *Electronics*, vol. 8(8), 2019. doi: 10.3390/electronics8080832.

[9] Cura T.: A particle swarm optimization approach to clustering, *Expert Systems with Applications*, vol. 39(1), pp. 1582–1588, 2012. doi: 10.1016/j.eswa.2011.07.123.

[10] Dafir Z., Lamari Y., Slaoui S.C.: A survey on parallel clustering algorithms for big data, *Artificial Intelligence Review*, vol. 54, pp. 2411–2443, 2021.

[11] Dasgupta S., Nave Frost M.M., Rashtchian C.: Explainable k-Means and k-Medians Clustering. In: *ICML'20: Proceedings of the 37th International Conference on Machine Learning*, pp. 7055–7065, 2020.

[12] Dean J., Ghemawat S.: MapReduce: simplified data processing on large clusters, *Communications of the ACM*, vol. 51(1), pp. 107–113, 2008. doi: 10.1145/1327452.1327492.

[13] Dhanabal L., Shantharajah S.P.: A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4(6), pp. 446–452, 2015.

[14] Guan Y., Ghorbani A.A., Belacel N.: Y-means: A clustering method for intrusion detection. In: *CCECE 2003 – Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No. 03CH37436)*, vol. 2, pp. 1083–1086, IEEE, 2003.

[15] HajKacem M.A.B., Moslah M., Essoussi N.: Spark Based Intrusion Detection System Using Practical Swarm Optimization Clustering. In: *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*, pp. 197–216, Springer, 2021. doi: 10.1007/978-3-030-74575-2_11.

[16] Horel E., Giesecke K., Storchan V., Chittar N.: Explainable clustering and application to wealth management compliance. In: *Proceedings of the First ACM International Conference on AI in Finance*, pp. 1–6, 2020. doi: 10.1145/3383455.3422530.

[17] Kennedy J., Eberhart R.: Particle swarm optimization. In: *Proceedings of ICNN'95 – International Conference on Neural Networks, Perth, WA, Australia, 1995*, vol. 4, pp. 1942–1948, IEEE, 1995. doi: 10.1109/ICNN.1995.488968.

[18] Keshk M., Koroniotis N., Pham N., Moustafa N., Turnbull B., Zomaya A.Y.: An explainable deep learning-enabled intrusion detection framework in IoT networks, *Information Sciences*, vol. 639, 2023. doi: 10.1016/j.ins.2023.119000.

[19] Li Z., Li Y., Xu L.: Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization. In: *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, vol. 2, pp. 157–161, IEEE, 2011. doi: 10.1109/icm.2011.184.

[20] Liao H.J., Lin C.H.R., Lin Y.C., Tung K.Y.: Intrusion detection system: A comprehensive review, *Journal of Network and Computer Applications*, vol. 36(1), pp. 16–24, 2013. doi: 10.1016/j.jnca.2012.09.004.

[21] Linardatos P., Papastefanopoulos V., Kotsiantis S.: Explainable AI: A review of machine learning interpretability methods, *Entropy*, vol. 23(1), 18, 2020. doi: 10.3390/e23010018.

[22] Liu H., Lang B.: Machine learning and deep learning methods for intrusion detection systems: A survey, *Applied Sciences*, vol. 9(20), 4396, 2019. doi: 10.3390/app9204396.

[23] Liu Y., Chen K., Liao X., Zhang W.: A genetic clustering method for intrusion detection, *Pattern Recognition*, vol. 37(5), pp. 927–942, 2004. doi: 10.1016/j.patcog.2003.09.011.

[24] Lundberg S.M., Lee S.I.: A Unified Approach to Interpreting Model Predictions. In: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds.), *Proceedings of 31st Conference on Neural Information Processing Systems (NIPS 2017), Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[25] MacQueen J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pp. 281–297, 1967.

[26] Madhulatha T.S.: An overview on clustering methods, *IOSR Journal of Engineering*, vol. 2(4), pp. 719–725, 2012. doi: 10.9790/3021-0204719725.

[27] Merwe van der D.W., Engelbrecht A.P.: Data clustering using particle swarm optimization. In: *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, vol. 1, pp. 215–220, IEEE, 2003.

[28] Moore J.D., Swartout W.R.: Explanation in expert systems: A survey, Technical report,USC University of Southern California, Marina del Rey Information Sciences Institute, 1988.

[29] Morichetta A., Casas P., Mellia M.: EXPLAIN-IT: Towards Explainable AI for Unsupervised Network Traffic Analysis. In: *Big-DAMA '19: Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, pp. 22–28, Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3359992.3366639.

[30] Moslah M., HajKacem M.A.B., Essoussi N.: Spark-Based Design of Clustering Using Particle Swarm Optimization. In: O. Nasraoui, C.E. Ben N'Cir (eds.), *Clustering Methods for Big Data Analytics: Techniques, Toolboxes and Applications*, pp. 91–113, Springer, Cham, 2019. doi: 10.1007/978-3-319-97864-2_5.

[31] Nanda S.J., Panda G.: A survey on nature inspired metaheuristic algorithms for partitional clustering, *Swarm and Evolutionary Computation*, vol. 16, pp. 1–18, 2014. doi: 10.1016/j.swevo.2013.11.003.

[32] Neupane S., Ables J., Anderson W., Mittal S., Rahimi S., Banicescu I., Seale M.: Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities, *IEEE Access*, vol. 10, pp. 112392–112415, 2022. doi: 10.1109/access.2022.3216617.

[33] Peng K., Leung V.C., Huang Q.: Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System Over Big Data, *IEEE Access*, vol. 6, pp. 11897–11906, 2018. doi: 10.1109/access.2018.2810267.

[34] Ribeiro M.T., Singh S., Guestrin C.: "Why should I trust you?": Explaining the predictions of any classifier. In: *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016. doi: 10.1145/2939672.2939778.

[35] Sculley D.: Web-scale k-means clustering. In: *WWW '10: Proceedings of the 19th international conference on World wide web*, pp. 1177–1178, 2010. doi: 10.1145/1772690.1772862.

[36] Shinde P.P., Shah S.: A Review of Machine Learning and Deep Learning Applications. In: *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India*, pp. 1–6, IEEE, 2018. doi: 10.1109/iccubea.2018.8697857.

[37] Tsai C.F., Hsu Y.F., Lin C.Y., Lin W.Y.: Intrusion detection by machine learning: A review, *Expert Systems with Applications*, vol. 36(10), pp. 11994–12000, 2009. doi: 10.1016/j.eswa.2009.05.029.

[38] Wang G., Hao J., Ma J., Huang L.: A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, *Expert Systems with Applications*, vol. 37(9), pp. 6225–6232, 2010. doi: 10.1016/j.eswa.2010.02.102.

[39] Wang M., Zheng K., Yang Y., Wang X.: An explainable machine learning framework for intrusion detection systems, *IEEE Access*, vol. 8, pp. 73127–73141, 2020. doi: 10.1109/access.2020.2988359.

[40] White T.: *Hadoop: The definitive guide*, O'Reilly Media, Inc., 3rd ed., 2012.

[41] Wu W., Xu S.: Intrusion Detection Based on Dynamic Gemini Population DE-K-mediods Clustering on Hadoop Platform, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 35(01), 2150001, 2021. doi: 10.1142/S0218001421500014.

[42] Xu Y., Qu W., Li Z., Min G., Li K., Liu Z.: Efficient k-Means++ Approximation with MapReduce, *IEEE Transactions on Parallel and Distributed Systems*, vol. 25(12), pp. 3135–3144, 2014.

[43] Younisse R., Ahmad A., Abu Al-Haija Q.: Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP), *Big Data and Cognitive Computing*, vol. 6(4), 126, 2022. doi: 10.3390/bdcc6040126.

[44] Zaharia M., Xin R.S., Wendell P., Das T., Armbrust M., Dave A., Meng X., *et al.*: Apache spark: a unified engine for big data processing, *Communications of the ACM*, vol. 59(11), pp. 56–65, 2016. doi: 10.1145/2934664.

[45] Zhao W., Ma H., He Q.: Parallel K-Means Clustering Based on MapReduce. In: M.G. Jaatun, G. Zhao, C. Rong (eds.), *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1–4, 2009. Proceedings 1*, pp. 674–679, Springer, 2009. doi: 10.1007/978-3-642-10665-1_71.

[46] Zhu W., Zeng N., Wang N.: Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS Implementations, *NESUG Proceedings: Health Care and Life Sciences, Baltimore, Maryland*, vol. 19, 67, 2010. https://lexjansen.com/nesug/nesug10/hl/hl07.pdf.

Affiliations

Chiheb Eddine Ben Ncir
University of Jeddah, College of Business, Saudi Arabia, cbenncir@uj.edu.sa

Mohamed Aymen Ben Haj Kacem
University of Tunis, LARODEC Laboratory, Tunisia, medaymenhajkacem@gmail.com

Mohammed Alattas
University of Jeddah, College of Business, Saudi Arabia, mialatas@uj.edu.sa

Received: 11.11.2023
Revised: 22.01.2024
Accepted: 22.01.2024

Ihor Yakymenko
Mykhailo Kasianchuk
Mikolaj Karpinski
Ruslan Shevchuk
Inna Shylinska

FINDING THE INVERSE OF A POLYNOMIAL MODULO IN THE RING Z[X] BASED ON THE METHOD OF UNDETERMINED COEFFICIENTS

Abstract *This paper presents the theoretical foundations of finding the inverse of a polynomial modulo in the ring $Z[x]$ based on the method of undetermined coefficients. The use of the latter makes it possible to significantly reduce the time complexity of calculations avoiding the operation of finding the greatest common divisor. An example of calculating the inverse of a polynomial modulo in the ring $Z[x]$ based on the proposed approach is given. Analytical expressions of the time complexities of the developed and classical methods depending on the degrees of polynomials are built. The graphic dependence of the complexity of performing the operation of finding the inverse of a polynomial in the ring $Z[x]$ is presented, which shows the advantages of the method based on undetermined coefficients. It is found that the efficiency of the developed method increases logarithmically with an increase in the degrees of polynomials.*

Keywords inverse of a polynomial modulo, ring of polynomials, Euclidean algorithm, degree of a polynomial, method of undetermined coefficients, time complexity, efficiency

Citation Computer Science 25(2) 2024: 239–252

1. Introduction

One of the most important and at the same time complex operation in the theory of algebra [1] and other mathematical applications [32] is the operation of finding the multiplicative inverse of a polynomial modulo in the ring $Z[x]$ [22,27]. The widespread use of this operation can be explained by its application in modern polynomial symmetric and asymmetric cryptography [24, 43] (in particular, post-quantum [11]), for solving problems of biometric identification of a person [3], in parallel and distributed calculations based on the Residue Number System [21, 34] in the ring $Z[x]$ [8, 36], data coding based on polynomial modular arithmetic [6, 38], signal and image processing [9, 17], for solving certain problems of linear programming [4] and in other applications of applied and discrete mathematics [25, 32, 41].

The methods of finding the multiplicative inverse of a polynomial in the ring $Z[x]$, as well as in the ring of integers [18, 20, 39], are characterized by significant computational complexities. Therefore, the development of new methods and improvement of the existing techniques for calculating the multiplicative inverse of a polynomial modulo in the ring $Z[x]$ is one of the most relevant problems.

One of the ways to solve this problem is the use of the method of undetermined coefficients. Nowadays, it is being successfully used for solving algebraic equations by factoring [10], decomposing a fraction, in which the numerator and denominator are polynomials, into simple fractions [5], finding the most optimal possible partial solutions to certain types of inhomogeneous ordinary differential equations [26], building some recurrent sequences [40], transforming logical functions, in particular, in Zhegalkin's algorithm [30], in integral calculus [29], in number methods [12].

According to the mentioned above, the purpose of our work is to develop a method for finding the inverse of a polynomial modulo based on the method of undetermined coefficients.

2. Related work

One of the most common methods for finding the multiplicative inverse of a polynomial in the ring $Z[x]$ is based on the extended Euclidean algorithm [7]. It should be noted that its application requires performing a large number of arithmetic operations on polynomials: division, finding residues [16], exponentiation [23], multiplication and substitution [14,44]. At the same time this method is characterized by the lowest time complexity compared to other known methods [19, 33].

When applying the Euclidean algorithm, finding the inverse of a polynomial is reduced to solving two problems: finding the greatest common divisor (GCD) of polynomials [28] and solving Diophantine equations [42] based on the extended Euclidean algorithm.

Let $f(x)$ and $g(x)$ be polynomials in the ring $Z[x]$ and $\deg(f(x)) > \deg(g(x))$, where the function *deg* denotes the degree of a polynomial. According to the main theorem of algebra for polynomials [31], there is a pair $q(x)$ and $r(x)$ from a ring $Z[x]$, for which $f(x) = q(x)g(x) + r(x)$, $0 < \deg r(x) < \deg g(x)$. Then, under the condition that $g(x)$ is not divisible by $r(x)$, for $g(x)$ the following equality is performed:

$$g(x) = r(x)q_1(x) + r_1(x), 0 < \deg r_1(x) < \deg r(x) \tag{1}$$

Next, if $r(x)$ is not divisible by $r_1(x)$, then:

$$r(x) = r_1(x)q_1(x) + r_2(x), 0 < \deg r_2(x) < \deg r_2(x) \tag{2}$$

This process is finite, that is, there exists such n, for which $r_{n-1}(x)$ will be divisible by $r_n(x)$.

As a result, a system of equations is obtained, on the basis of which, the GCD of two polynomials is found:

$$\begin{gathered}
f(x) = q(x)g(x) + r(x), 0 < \deg r(x) < \deg g(x)\\
g(x) = r(x)q_1(x) + r_1(x), 0 < \deg r_1(x) < \deg r(x)\\
r(x) = r_1(x)q_2(x) + r_2(x), 0 < \deg r_2(x) < \deg r_1(x)\\
\dots\dots\dots\dots\dots\\
r_{n-2}(x) = r_{n-1}(x)q_{n-1}(x) + r_n(x), 0 < \deg r_n(x) < \deg r_{n-1}(x)\\
r_{n-1}(x) = r_n(x)q_n(x)
\end{gathered} \tag{3}$$

Sequence (3) determines the steps of applying the Euclidean algorithm, according to which the relationship of the GCD of polynomials comes true:

$$\begin{gathered}
GCD(f(x), g(x)) = GCD(g(x), r(x)) = GCD(r(x), r_1(x)) = \dots\\
= GCD(r_{n-1}(x), r_n(x)) = r_n(x)
\end{gathered} \tag{4}$$

Calculation of the inverse of a polynomial in the ring $Z[x]$ is reduced to solving the Diophantine equation, since the two relatively prime polynomials $f(x)$ and $g(x)$ can match the following polynomials $l(x)$, $s(x) \in Z[x]$ for which the equality $f(x) \cdot l(x) + g(x) \cdot s(x) = GCD(f(x), g(x))$=$w$ holds. If $f(x)$ and $g(x)$ are not relatively prime polynomials, then according to the definition of the ring, the inverse of a polynomial does not exist.

To simplify the procedure for finding the inverse of a polynomial, formula (4) must be written as follows:

$$\begin{gathered}
r(x) = f(x) - q(x)g(x)\\
r_1(x) = g(x) - r(x)q_1(x) = g(x) - (f(x) - q(x)g(x))q_1(x) =\\
g(x)(1 + q(x)q_1(x)) - f(x)q_1(x)\\
\dots\dots\dots\dots\dots\\
GCD(f(x), g(x)) = r_n(x) = f(x)l(x) + g(x)s(x)
\end{gathered} \tag{5}$$

The notation $GCD(f(x), g(x))$ is called the Bezout relation for polynomials and the polynomials $l(x)$ and $s(x)$ are Bezout's polynomials. Then, $f(x)^{-1} \bmod g(x) = l(x) \bmod g(x)$.

In [45], the mathematical foundations of permutation polynomials, whose degrees do not exceed 6, and their inverse polynomials in finite fields were presented. It was noted that the results of the conducted research could be applied in cryptography, coding theory and combinatorial design theory.

Work [35] is devoted to the problem of finding the inverse of a polynomial in the Nth Degree Truncated Polynomial Ring. The method proposed in this work is based on finding the inverse of the polynomial using an adaptive inverse algorithm determined in the field of polynomials with binary and ternary coefficients. It was also noted that this algorithm could be extended for polynomials with coefficients of different fields, including $Z(x)$. The efficiency of the algorithm was studied in comparison with the inverse algorithm of Zhao and Su.

In [25], the method for inverse polynomial mappings was used to build the optimal interpolation nodes on discrete intervals. It was shown that this method was highly efficient for T-polynomials. In [13], it was noted that there was a length limitation for the inverse polynomial depending on the value of the acceptable error. In addition, five examples of polynomial inversion for solving physics and mathematics problems were presented. In [37], an algorithm generating the inverse elements over finite fields GF (3^m) was presented. Calculations were based on multiplication, squaring and cubing.

3. Method for calculating the inverse of a polynomial modulo in the ring Z(x)

Let us find the inverse of a polynomial modulo $m(x) = r^{-1}(x) \bmod g(x)$ in the ring of polynomials $Z(x)$, where $r(x)$ and $g(x)$ are relatively prime polynomials $(GCD(r(x), g(x)) = w, w \in Z)$ and $\deg r(x) = n$, $\deg g(x) = l$ for which the equality holds:

$$r(x) \cdot m(x) \equiv w \bmod g(x) \tag{6}$$

At the same time, the degree of a polynomial $\deg m(x) = l - 1$, since the residue modulo $g(x)$ will be a polynomial of $l-1$ degree. Let $f(x) = r(x) \cdot m(x)$ and $r(x) = A_n x^n + A_{n-1}x^{n-1} + \ldots + A_1 x + A_0$, $g(x) = B_l x^l + B_{l-1}x^{l-1} + \ldots + B_1 x + B_0$, then $m(x) = C_k x^k + C_{k-1}x^{k-1} + \ldots + C_1 x + C_0$, where $A_i, B_j, C_k \in Z$, $i = 0, \ldots, n$, $j = 0, \ldots, l, k = 0, \ldots, l-1$. Based on the method of undetermined coefficients, equality (6) for polynomials $r(x)$, $g(x)$ and $m(x)$ can be written as follows:

$$\begin{aligned} &x^n + A_{n-1}x^{n-1} + \ldots + A_1 x + A_0) \cdot (C_k x^k + C_{k-1}x^{k-1} + \ldots \\ &+ C_1 x + C_0)) \bmod (B_l x^l + B_{l-1}x^{l-1} + \ldots + B_1 x^1 + B_0) = w \end{aligned} \tag{7}$$

or

$$\begin{aligned} &((A_n x^n + A_{n-1}x^{n-1} + \ldots + A_1 x + A_0) \cdot (C_k x^k + C_{k-1}x^{k-1} + \ldots \\ &+ C_1 x + C_0)) \bmod (B_l x^l + +B_{l-1}x^{l-1} + \ldots + B_1 x^1 + B_0) - w = 0 \end{aligned} \tag{8}$$

First, we need to find the value $f(x) = r(x) \cdot m(x)$:

$$\begin{gathered} f(x) = A_n x^n \cdot (C_k x^k + C_{k-1} x^{k-1} + \ldots \\ + C_1 x + C_0) + A_{n-1} x^{n-1} (C_k x^k + C_{k-1} x^{k-1} + \ldots \\ + C_1 x + C_0) + \ldots + + A_1 x (C_k x^k + C_{k-1} x^{k-1} + \ldots \\ + C_1 x + C_0) + A_0 (C_k x^k + C_{k-1} x^{k-1} + \ldots + C_1 x + C_0) = \\ A_n C_k x^{n+k} + (A_n C_{k-1} + A_{n-1} C_k) x^{n+k-1} + \\ (A_n C_{k-2} + A_{n-1} C_{k-1} + A_{n-2} C_k) x^{n+k-2} + \ldots \\ + (A_0 C_1 + A_1 C_0) x + A_0 C_0 \end{gathered} \tag{9}$$

In expression (9), $\deg f(x) = n + k$. Let us introduce the notations: $F_{n+k} = A_n C_k$, $F_{n+k-1} = (A_n C_{k-1} + A_{n-1} C_k)$, $F_{n+k-2} = (A_n C_{k-2} + A_{n-1} C_{k-1} + A_{n-1} C_k)$, $\ldots$, $F_1 = (A_0 C_1 + A_1 C_0)$,$F_0 = A_0 C_0$. Then, equality (9) can be written as follows:

$$f(x) = F_{n+k} x^{n+k} + F_{n+k-1} x^{n+k-1} + \ldots + F_1 x + F_0 \tag{10}$$

The problem of determining the coefficients $C_k \in Z$ arises, for which condition (7) is fulfilled. Therefore, it is necessary to find the residue on division $f(x)$ by $g(x)$:

$$\begin{gathered} (F_{n+k} x^{n+k} + F_{n+k-1} x^{n+k-1} + \ldots \\ + F_1 x + F_0) \bmod (B_l x^l + B_{l-1} x^{l-1} + \ldots + B_1 x^1 + B_0) \end{gathered} \tag{11}$$

Let us consider the polynomial:

$$f(x) - \frac{F_{n+k}}{B_l} x^{n+k-l} g(x) = f_1(x), B_l \neq 0 \tag{12}$$

Given (10), expression (12) can be written as follows:

$$\begin{gathered} (F_{n+k} x^{n+k} + F_{n+k-1} x^{n+k-1} + \ldots \\ + F_1 x + F_0) - \frac{F_{n+k}}{B_l} x^{n+k-l} (B_l x^l + B_{l-1} x^{l-1} + \ldots \\ + B_1 x^1 + B_0) = f_1(x) \end{gathered} \tag{13}$$

Moreover, $\deg f(x) > \deg f_1(x)$ and the coefficient of the highest degree is determined according to the relation: $f_1(x) = (\frac{F_{n+k-1} \cdot B_l - F_{n+k} \cdot B_{l-1}}{B_l^2}) = F_{1nk}$, $\deg f_1(x) = n + k - 1 = n_1$.

If $\deg f_1(x) > \deg g(x)$, then the equality can be written as follows:

$$\begin{gathered} f_1(x) - (\frac{F_{n+k-1} \cdot B_l - F_{n+k} \cdot B_{l-1}}{B_l^2}) x^{n+k-l-1} g(x) = \\ f_1(x) - F_{1nk} x^{n+k-l-1} g(x) = f_2(x) \end{gathered} \tag{14}$$

where $\deg f_1(x) > \deg f_2(x)$, coefficient $f_2(x) = F_{2n}$ and $\deg f_2(x) = n_2$. It is not difficult to make sure that$\deg f_2(x) > \deg g(x)$, that is $n_2 \geq l$. Therefore, you can continue this procedure and write the following equality:

$$f_2(x) - \frac{F_{2nk}}{B_l} x^{n_2-l} g(x) = f_3(x) \tag{15}$$

where the condition $\deg f_2(x) > \deg f_3(x)$ is fulfilled for degrees of polynomials, the coefficient of the maximum degree of the polynomial $f_3(x)$ is F_{3nk} and $\deg f_3(x) = n_3$.

If $n_3 \geq l$, then the following equality is obtained in a similar way:

$$f_3(x) - \frac{F_{3nk}}{B_l} x^{n_3-l} g(x) = f_4(x) \tag{16}$$

where $\deg f_2(x) > \deg f_3(x)$. According to the above notations, $f_3(x) = F_{3nk}$ and $\deg f_3(x) = n_3$. It should be noted that the degrees of the created polynomials $f_1(x)$, $f_2(x)$, $f_3(x)$, $\ldots$ decrease ($n_1 > n_2 > n_3 > \ldots$), therefore, after a finite number of steps s the following polynomial is obtained:

$$f_s(x) - \frac{F_{snk}}{B_l} x^{n_s-l} g(x) = f_{s+1}(x) \tag{17}$$

The degrees of polynomials satisfy the inequality $\deg f_s(x) > \deg f_{s+1}(x)$. According to the above relationships, the value of the residue is as follows: $(F_{n+k}x^{n+k} + F_{n+k-1}x^{n+k-1} + \ldots + F_1 x + F_0) \bmod (B_l x^l + B_{l-1}x^{l-1} + B_1 x^1 + B_0) = f_{s+1}(x)$.

This leads to the following relationship adding all the Equations (14)–(17):

$$\begin{gathered} f(x) - \frac{F_{n+k}}{B_l} x^{n+k-l} g(x) + f_1(x) - \\ \frac{F_{1nk}}{B_l} x^{n_1-l} g(x) + f_2(x) - \frac{F_{2nk}}{B_l} x^{n_2-l} g(x) + f_3(x) - \\ \frac{F_{3nk}}{B_l} x^{n_3-l} g(x) + \ldots + f_s(x) - \frac{F_{snk}}{B_l} x^{n_s-l} g(x) = \\ (f_1(x) + f_2(x) + f_3(x) + \ldots + f_s(x)) \Rightarrow \\ f(x) - (\frac{F_{n+k}}{B_l} x^{n+k-l} + \frac{F_{1nk}}{B_l} x^{n_1-l} + \\ \frac{F_{2nk}}{B_l} x^{n_2-l} + \frac{F_{3nk}}{B_l} x^{n_3-l} + \ldots + \frac{F_{snk}}{B_l} x^{n_s-l}) = f_{s+1}(x) \end{gathered} \tag{18}$$

As a result of transformations (18), a polynomial $f_{s+1}(x) = L_{s-1}x^{s-1} + L_{s-2}x^{s-2} + \ldots + L_1 x + L_0$ of $\deg f_{s+1}(x) = s-1$ order is obtained. Taking into account condition (7), the method of undetermined coefficients for calculating values $C_k \in Z$, where $k = 1, \ldots, l-1$, leads to a system of s equations and s unknowns, which must be found:

$$L_{s-1} = 0,\ L_{s-2} = 0, \ldots, L_1 = 0,\ L_0 = w \tag{19}$$

According to these equations, the value $C_k \in Z$ is calculated. The scheme of finding the inverse of a polynomial in a ring $Z(x)$ based on the method of undetermined coefficients is presented in Figure 1.

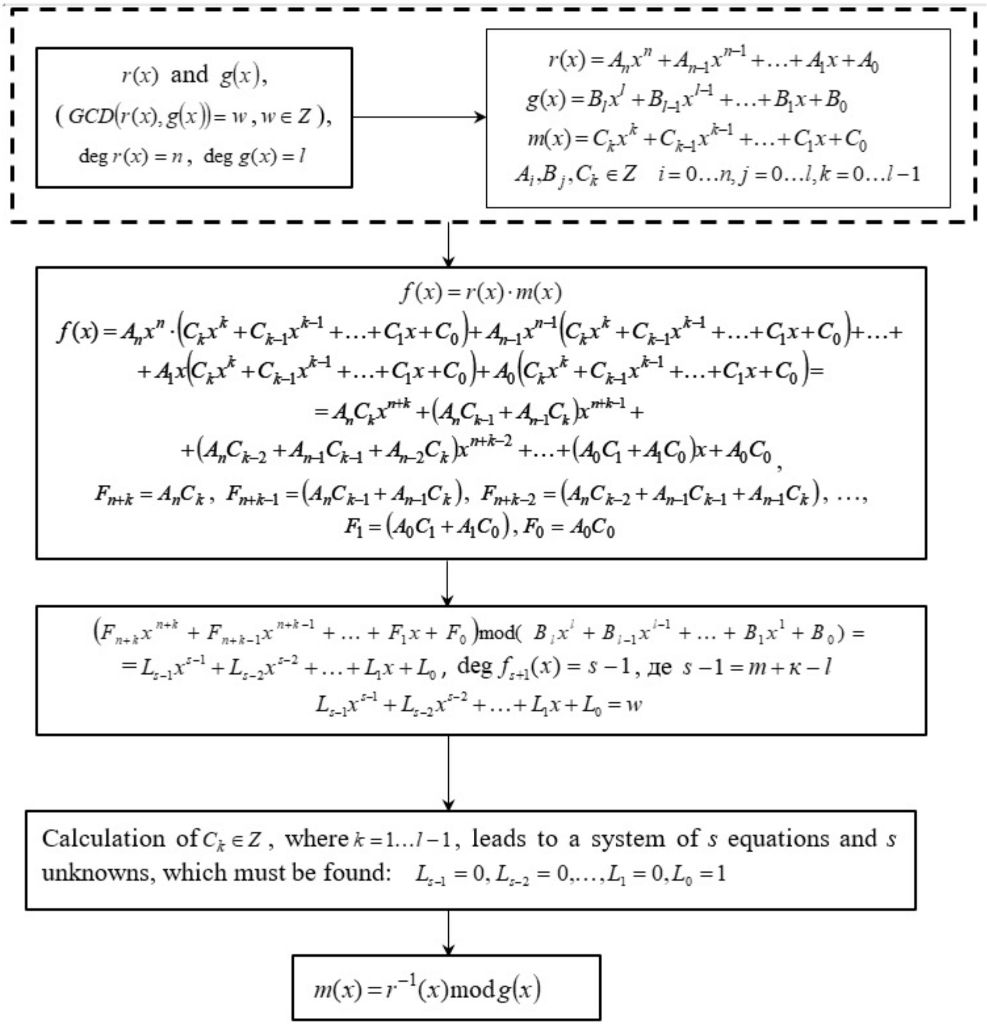

$r(x)$ and $g(x)$,
($GCD(r(x), g(x)) = w$, $w \in Z$),
$\deg r(x) = n$, $\deg g(x) = l$
$r(x) = A_nx^n + A_{n-1}x^{n-1} + \ldots + A_1x + A_0$
$g(x) = B_lx^l + B_{l-1}x^{l-1} + \ldots + B_1x + B_0$
$m(x) = C_kx^k + C_{k-1}x^{k-1} + \ldots + C_1x + C_0$
$A_i, B_j, C_k \in Z \quad i = 0\ldots n, j = 0\ldots l, k = 0\ldots l-1$
$f(x) = r(x) \cdot m(x)$
$f(x) = A_nx^n \cdot (C_kx^k + C_{k-1}x^{k-1} + \ldots + C_1x + C_0) + A_{n-1}x^{n-1}(C_kx^k + C_{k-1}x^{k-1} + \ldots + C_1x + C_0) + \ldots +$
$+ A_1x(C_kx^k + C_{k-1}x^{k-1} + \ldots + C_1x + C_0) + A_0(C_kx^k + C_{k-1}x^{k-1} + \ldots + C_1x + C_0) =$
$= A_nC_kx^{n+k} + (A_nC_{k-1} + A_{n-1}C_k)x^{n+k-1} +$
$+ (A_nC_{k-2} + A_{n-1}C_{k-1} + A_{n-2}C_k)x^{n+k-2} + \ldots + (A_0C_1 + A_1C_0)x + A_0C_0$,
$F_{n+k} = A_nC_k$, $F_{n+k-1} = (A_nC_{k-1} + A_{n-1}C_k)$, $F_{n+k-2} = (A_nC_{k-2} + A_{n-1}C_{k-1} + A_{n-1}C_k)$, ...,
$F_1 = (A_0C_1 + A_1C_0)$, $F_0 = A_0C_0$
$(F_{n+k}x^{n+k} + F_{n+k-1}x^{n+k-1} + \ldots + F_1x + F_0)$mod($B_lx^l + B_{l-1}x^{l-1} + \ldots + B_1x^1 + B_0$) =
$= L_{s-1}x^{s-1} + L_{s-2}x^{s-2} + \ldots + L_1x + L_0$, $\deg f_{s+1}(x) = s-1$, де $s-1 = m + \kappa - l$
$L_{s-1}x^{s-1} + L_{s-2}x^{s-2} + \ldots + L_1x + L_0 = w$
Calculation of $C_k \in Z$, where $k = 1\ldots l-1$, leads to a system of s equations and s unknowns, which must be found: $L_{s-1} = 0, L_{s-2} = 0, \ldots, L_1 = 0, L_0 = 1$
$m(x) = r^{-1}(x) \bmod g(x)$

Figure 1. The scheme of finding the inverse of a polynomial in the polynomial ring

4. An example of the application of the developed method

Let the polynomials be $r(x) = x^2 + 3x + 1$ and $g(x) = x^3 + 3x^2 + 2x + 1$. It is necessary to find $m(x) = r(x)^{-1} \bmod g(x) = (x^2+3x+1)^{-1} \bmod (x^3+3x^2+2x+1)$. According to the theoretical provisions presented above, the polynomial $m(x)$ is as follows: $m(x) = Ax^2 + Bx + C$. Taking into account (9), it is possible to obtain the product: $r(x) \cdot m(x) = (x^2 + 3x + 1) \cdot (Ax^2 + Bx + C) = Ax^4 + 3Ax^3 + Ax^2 + Bx^3 + 3Bx^2 + Bx + Cx^2 + 3Cx + C = Ax^4 + (3A+B)x^3 + (A+3B+C)x^2 + (B+3C)x + C$. Next, using the relationships (14)–(17), the residue of the division $r(x) \cdot m(x) \bmod g(x) = (Ax^4 + (3A + B)x^3 + (A + 3B + C)x^2 + (B + 3C)x + C) \bmod (x^3 + 3x^2 + 2x + 1)$ is obtained: $Ax^4 + (3A + B)x^3 + (2A + 3B)x^2 + (A + 2B)x + B = (Ax + B) \cdot (x^3 + 3x^2 + 2x + 1) + ((-A + C)x^2 + (-A - B + 3C)x + C - B)$.

As a result, a polynomial $f_3(x) = (-A + C)x^2 + (-A - B + 3C)x + C - B$, whose degree is less than deg $g(x)$, is obtained as a residue. To simplify the calculations, let $f_3(x) = w = 1$, that, $(-A + C)x^2 + (-A - B + 3C)x + C - B = 1$. Then condition (19) for finding the unknown coefficients A, B and C leads to the following system of equations:

$$\begin{cases} C - A = 0 \\ 3C - A - B = 0 \\ C - B = 1 \end{cases} \tag{20}$$

Its solution determines the coefficients: $A = -1, B = -2, C = -1$.

Thus, the value of the inverse of a polynomial modulo in the ring Z[x] is calculated as follows: $m(x) = r(x)^{-1} \bmod g(x) = (x^2 + 3x + 1)^{-1} \bmod (x^3 + 3x^2 + 2x + 1) = -x^2 - 2x - 1$.

5. Estimating the computational complexity of the proposed algorithm for calculating the inverse of a polynomial in the ring Z[X]

When building analytical expressions for estimating the time complexity of calculating the inverse in a ring of polynomials according to the classical and proposed methods, it is necessary to determine the complexity of the most time-consuming operations, namely:

1. Product of two polynomials.
2. The residue of two polynomials.

At the first step of the implementation of the proposed method, according to (9), the most computationally complex operation is the multiplication of two polynomials $(A_n x^n + A_{n-1} x^{n-1} + \ldots + A_1 x + A_0) \cdot (C_k x^k + C_{k-1} x^{k-1} + \ldots + C_1 x + C_0)$. Its time complexity for polynomials of n degree was studied in [15] and consists of $O(n \log n)$ bit operations, where the logarithm is taken to the base 2. Given the complexity of finding residues [15], the general estimate is $O(2n \log n)$ of bit operations.

A well-known method of finding the inverse of a polynomial in a polynomial ring is based on the use of the Euclidean algorithm and its consequences. In [2], it was noted that the time complexity of finding the GCD $(p(x),q(x))$ over the field $Z[X]$ according to the Euclidean algorithm has an upper limit $O(n \log^2 n)$, where

$n = \max\{\deg(p), \deg(q)\}$. In addition, the best known asymptotic estimate of the Euclidean inverse algorithm is equal to $O(n \log n \log\log n) \approx O(n \log n)$ [2].

Table 1 shows the basic operations and time complexities when using the proposed method for finding the inverse of a polynomial based on the method of undetermined coefficients and the classical method based on the extended Euclidean algorithm.

Table 1

Basic operations and time complexities of the proposed and classical methods for finding the inverse of a polynomial

Basic operations	Time complexity of the classical method, $O(n)$	Time complexity of the proposed method, $O(n)$
Computing the GCD (a, b) over the field by the Euclidean algorithm	$O(n \log^2 n)$, where $n = \max\{\deg(p), \deg(q)\}$	–
The extended Euclidean algorithm	$O(n \log n \log\log n) \approx O(n \log n)$	–
The product of two polynomials	–	$O(n \log n)$
Finding the residue of two polynomials	–	$O(n \log n)$

Taking into account the time complexities of the basic operations, the overall time complexity of the classical method of finding the inverse in the polynomial ring is $O1(n \log n \cdot (1 + \log n))$. Since, using the proposed method you do not need to find the GCD of polynomials, then the time complexity will decrease: $O2(2n \log n)$. Figure 2 shows the graphs that characterize the dependences of the time complexities of the proposed and classical methods for finding the inverse of a polynomial in the polynomial ring on the polynomial degrees.

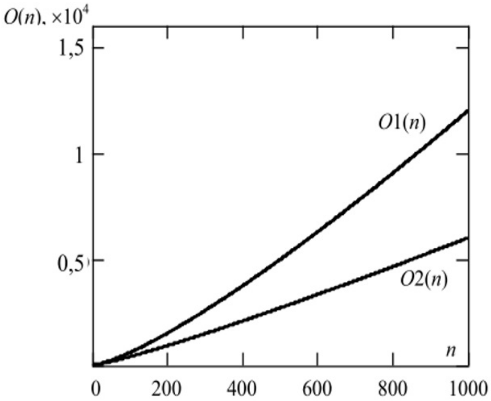


Figure 2. Time complexities of the classical $O1(n)$ and proposed $O2(n)$ methods

As a result of the numerical experiment, it was found that the complexity increases significantly with an increase in the degrees of polynomials.

The efficiency of the developed method in comparison with the classical one is defined as the relation of the respective complexities:

$$E(n) = \frac{O1(n)}{O2(n)} = \frac{n \log n \cdot (1 + \log n)}{2n \log n} = \frac{1 + \log n}{2} \tag{21}$$

Due to expression (21), the found efficiency increases logarithmically with an increase in degrees of polynomials.

6. Conclusions

For the first time, an algorithm for finding the inverse of a polynomial in the ring $Z[x]$ based on the method of undetermined coefficients is proposed. A mathematical description of the developed method is presented and an example of its application is given. Analytical expressions of time complexities are built depending on the order of polynomials for the proposed method and the known one based on the Euclidean algorithm and its consequences. As a result of the conducted research, the higher efficiency of the algorithm based on the method of undetermined coefficients without finding the GCD of polynomials, has been proven. One of the main advantages of the proposed approach is the reduction of the time complexity from $O1(n \log n \cdot (1 + \log n))$ to $O2(2n \log n)$ compared to the known method. Graphic dependences of time complexities are presented. It is found that the efficiency of the proposed method increases logarithmically with an increase in the degrees of polynomials.

Further research in this field can be devoted to the development of the method for a polynomial recovery from its residues (Chinese Remainder Theorem for Polynomials), development of the theoretical foundations of the Residue Number System in the ring $Z[x]$, its perfect and modified perfect forms, as well as the development of new polynomial encryption algorithms with increased resistance to cryptanalysis. In addition, we are currently working on the software and hardware implementation of the proposed algorithm, which will make it possible to detect limitations of the usage of some polynomial classes in the method of undetermined coefficients.

References

[1] Abdulazeez S.T., Hussein A.M.: The Existence of a Polynomial Inverse Integrating Factors and Studies About the Limit Cycles for Cubic, Quartic and Quintic Polynomial Systems, *Baghdad Science Journal*, vol. 18(2), pp. 0322–0322, 2021. doi: 10.21123/bsj.2021.18.2.0322.

[2] Aho A.V., Hopcroft J.E.: *The design and analysis of computer algorithms*, Pearson Education India, 1974.

[3] Andrijchuk V.A., Kuritnyk I.P., Kasyanchuk M.M., Karpinski M.P.: Modern algorithms and methods of the person biometric identification. In: *2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 403–406, IEEE, 2005. doi: 10.1109/idaacs.2005.283012.

[4] Ashraphijuo M., Madani R., Lavaei J.: Inverse function theorem for polynomial equations using semidefinite programming. In: *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 6589–6596, IEEE, 2015. doi: 10.1109/cdc.2015.7403257.

[5] Beyer W.H.: *CRC standard mathematical tables. 25th Edition*, CRC Press, West Palm Beach, 1978.

[6] Boucher D., Ulmer F.: Linear codes using skew polynomials with automorphisms and derivations, *Designs, Codes and Cryptography*, vol. 70, pp. 405–431, 2014. doi: 10.1007/s10623-012-9704-4.

[7] Calvez L.C., Azou S., Vilbé P.: Variation on Euclid's algorithm for polynomials, *Electronics Letters*, vol. 33(11), 1997. doi: 10.1049/el:19970658.

[8] Chu J., Benaissa M.: GF(2^{m}) multiplier using Polynomial Residue Number System. In: *APCCAS 2008 – 2008 IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1514–1517, IEEE, 2008. doi: 10.1109/APCCAS.2008.4746320.

[9] Dadkhahi H., Gotchev A., Egiazarian K.: Inverse polynomial reconstruction method in DCT domain, *EURASIP Journal on Advances in Signal Processing*, vol. 2012, pp. 1–23, 2012. doi: 10.1186/1687-6180-2012-133.

[10] De Leon D.: Using Undetermined Coefficients to Solve Certain Classes of Variable-Coefficient Equations, *The American Mathematical Monthly*, vol. 122(3), pp. 246–255, 2015. doi: 10.4169/amer.math.monthly.122.03.246.

[11] Drucker N., Gueron S., Kostic D.: Fast polynomial inversion for post quantum QC-MDPC cryptography, *Information and Computation*, vol. 281, 104799, 2021. doi: 10.1016/j.ic.2021.104799.

[12] Dubeau F.: The method of undetermined coefficients: general approach and optimal error bounds, *Journal of Mathematical Analysis*, vol. 5(4), pp. 1–11, 2014. http://www.ilirias.com/jma/repository/docs/JMA5-4-1.pdf.

[13] González-Cardel M.F., Díaz-Uribe R.: An analysis on the inversion of polynomials, *Revista Mexicana de Física*, vol. 52(2), pp. 163–171, 2006. https://rmf.smf.mx/ojs/index.php/rmf-e/article/view/4522.

[14] Goupil A., Palicot J.: Variation on variation on Euclid's algorithm, *IEEE Signal Processing Letters*, vol. 11(5), pp. 457–458, 2004. doi: 10.1109/lsp.2004.824053.

[15] Harvey D., Hoeven van der J.: Faster polynomial multiplication over finite fields using cyclotomic coefficient rings, *Journal of Complexity*, vol. 54, 101404, 2019. doi: 10.1016/j.jco.2019.03.004.

[16] Ivasiev S., Kasianchuk M.M., Yakymenko I.Z., Shevchuk R., Karpinski M., Gomotiuk O.: Effective algorithms for finding the remainder of multi-digit numbers. In: *2019 9th International Conference on Advanced Computer Information Technologies (ACIT)*, pp. 175–178, IEEE, 2019. doi: 10.1109/acitt.2019.8779899.

[17] Jassim W.A., Raveendran P., Mukundan R.: New orthogonal polynomials for speech signal and image processing, *IET Signal Processing*, vol. 6(8), pp. 713–723, 2012. doi: 10.1049/iet-spr.2011.0004.

[18] Karpinski M., Rajba S., Zawislak S., Warwas K., Kasianchuk M.M., Ivasiev S., Yakymenko I.: A Method for Decimal Number Recovery from its Residues Based on the Addition of the Product Modules. In: *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, pp. 13–17, IEEE, 2019. doi: 10.1109/idaacs.2019.8924395.

[19] Kasianchuk M., Yakymenko I., Pazdriy I., Zastavnyy O.: Algorithms of findings of perfect shape modules of remaining classes system. In: *The Experience of Designing and Application of CAD Systems in Microelectronics*, pp. 316–318, IEEE, 2015. doi: 10.1109/cadsm.2015.7230866.

[20] Kasianchuk M.M., Nykolaychuk Y.N., Yakymenko I.Z.: Theory and methods of constructing of modules system of the perfect modified form of the system of residual classes, *Journal of Automation and Information Sciences*, vol. 48(8), 2016. doi: 10.1615/jautomatinfscien.v48.i8.60.

[21] Kasianchuk M.M., Yakymenko I.Z., Nykolaychuk Y.M.: Symmetric Cryptoalgorithms in the Residue Number System, *Cybernetics and Systems Analysis*, vol. 57(2), pp. 329–336, 2021. doi: 10.1007/s10559-021-00358-6.

[22] Katagiri Y., Iwamura K., Nakanishi Y., Takano S., Suzuki R.: Arbitrary polynomial chaos expansion and its application to power flow analysis-Fast approximation of probability distribution by arbitrary polynomial expansion, *Journal of Physics: Conference Series*, vol. 1780, 012025, 2021. doi: 10.1088/1742-6596/1780/1/012025.

[23] Kozaczko D., Ivasiev S., Yakymenko I., Kasianchuk M.: Vector module exponential in the remaining classes system. In: *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, pp. 161–163, IEEE, 2015. doi: 10.1109/idaacs.2015.7340720.

[24] Krausz M., Land G., Richter-Brockmann J., Güneysu T.: Efficiently masking polynomial inversion at arbitrary order. In: *International Conference on Post-Quantum Cryptography*, pp. 309–326, Springer, 2022. doi: 10.1007/978-3-031-17234-2_15.

[25] Kroó A., Szabados J.: Inverse polynomial mappings and interpolation on several intervals, *Journal of Mathematical Analysis and Applications*, vol. 436(2), pp. 1165–1179, 2016. doi: 10.1016/j.jmaa.2015.12.032.

[26] Kwasniewski L.: An Improved Method of Undetermined Coefficients, *The Open Applied Mathematics Journal*, vol. 3(1), pp. 33–39, 2009. doi: 10.2174/1874114200903010033.

[27] Lasserre J.B.: Inverse polynomial optimization, *Mathematics of Operations Research*, vol. 38(3), pp. 418–436, 2013. doi: 10.1287/moor.1120.0578.

[28] Liu C.L., Horng G., Liu H.Y.: Computing the modular inverses is as simple as computing the GCDs, *Finite Fields and Their Applications*, vol. 14(1), pp. 65–75, 2008. doi: 10.1016/j.ffa.2007.08.004.

[29] Logan J.D.: *A First Course in Differential Equations*, Springer, New York, NY, 2006. doi: 10.1007/978-1-4419-7592-8.

[30] Lombardi H., Quitté C.: The Method of Undetermined Coefficients. In: *Commutative Algebra: Constructive Methods: Finite Projective Modules*, pp. 77–172, Springer, 2015. doi: 10.1007/978-94-017-9944-7_3.

[31] Milne J.S.: Algebraic Number Theory (v3.08), 2020, https://wwwjmilneorg/math/CourseNotes/ANTpdf.

[32] Moreno J., Saiz A.: Inverse functions of polynomials and its applications to initialize the search of solutions of polynomials and polynomial systems, *Numerical Algorithms*, vol. 58(2), pp. 203–233, 2011. doi: 10.1007/s11075-011-9453-x.

[33] Nykolaychuk Y.M., Kasianchuk M.M., Yakymenko I.Z.: Theoretical foundations for the analytical computation of coefficients of basic numbers of Krestenson's transformation, *Cybernetics and Systems Analysis*, vol. 50, pp. 649–654, 2014. doi: 10.1007/s10559-014-9654-0.

[34] Nykolaychuk Y.M., Yakymenko I.Z., Vozna N.Y., Kasianchuk M.M.: Residue Number System Asymmetric Cryptoalgorithms, *Cybernetics and Systems Analysis*, vol. 58(4), pp. 611–618, 2022. doi: 10.1007/s10559-022-00494-7.

[35] Nyokabi G.J., Salleh M., Mohamad I.: NTRU inverse polynomial algorithm based on circulant matrices using gauss-jordan elimination. In: *2017 6th ICT International Student Project Conference (ICT-ISPC)*, pp. 1–5, IEEE, 2017. doi: 10.1109/ict-ispc.2017.8075326.

[36] Paliouras V., Skavantzos A., Stouraitis T.: Low power convolvers using the Polynomial Residue Number System. In: *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No. 02CH37353)*, vol. 2, pp. II–II, IEEE, 2002.

[37] Park C.M.: A Study on Constructing the Inverse Element Generator over GF (3 m), *Journal of Information and Communication Convergence Engineering*, vol. 8(3), pp. 317–322, 2010.

[38] Puchinger S., Wachter-Zeh A.: Fast operations on linearized polynomials and their applications in coding theory, *Journal of Symbolic Computation*, vol. 89, pp. 194–215, 2018. doi: 10.1016/j.jsc.2017.11.012.

[39] Rajba T., Klos-Witkowska A., Ivasiev S., Yakymenko I.Z., Kasianchuk M.M.: Research of time characteristics of search methods of inverse element by the module. In: *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, pp. 82–85, IEEE, 2017. doi: 10.1109/idaacs.2017.8095054.

[40] Rosen K.H.: *Handbook of discrete and combinatorial mathematics*, CRC Press, 1999.

[41] Shams M., Rafiq N., Kausar N., Ahmed S.F., Mir N.A., Chandra Saha S.: Inverse Family of Numerical Methods for Approximating All Simple and Roots with Multiplicity of Nonlinear Polynomial Equations with Engineering Applications, *Mathematical Problems in Engineering*, vol. 2021, pp. 1–9, 2021. doi: 10.1155/2021/3124615.

[42] Xiao F., Lu D., Wang D.: Solving multivariate polynomial matrix Diophantine equations with Gröbner basis method, *Journal of Systems Science and Complexity*, vol. 35(1), pp. 413–426, 2022.

[43] Yakymenko I., Kasianchuk M., Shylinska I., Shevchuk R., Yatskiv V., Karpinski M.: Polynomial Rabin Cryptosystem Based on the Operation of Addition. In: *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)*, pp. 345–350, IEEE, 2022. doi: 10.1109/acit54803.2022.9913089.

[44] Yakymenko I., Kasyanchuk M., Nykolajchuk Y.: Matrix algorithms of processing of the information flow in computer systems based on theoretical and numerical Krestenson's basis. In: *2010 International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, pp. 241–241, IEEE, 2010.

[45] Zheng Y., Wang Q., Wei W.: On inverses of permutation polynomials of small degree over finite fields, *IEEE Transactions on Information Theory*, vol. 66(2), pp. 914–922, 2019. doi: 10.1109/tit.2019.2939113.

Affiliations

Ihor Yakymenko
West Ukrainian National University, Department of Cyber Security, 46009 Ternopil, Ukraine

Mykhailo Kasianchuk
West Ukrainian National University, Department of Cyber Security, 46009 Ternopil, Ukraine

Mikolaj Karpinski
University of the National Education Commission, Institute of Security and Computer Science, 30-084 Krakow, Poland
Ternopil Ivan Puluj National Technical University, Department of Cyber Security, 46001 Ternopil, Ukraine

Ruslan Shevchuk
University of Bielsko-Biala, Department of Computer Science and Automatics, 43-309 Bielsko-Biala, Poland
West Ukrainian National University, Department of Computer Science, 46009 Ternopil, Ukraine

Inna Shylinska
West Ukrainian National University, Foreign Languages and Information Communication Technologies Department, 46009 Ternopil, Ukraine

Received: 10.09.2023
Revised: 02.03.2024
Accepted: 21.03.2024

Nirupam Shome
Devran Dey Sarkar
Richik Kashyap
Rabul Hussain Laskar

DETECTION OF CREDIT CARD FRAUD WITH OPTIMIZED DEEP NEURAL NETWORK IN BALANCED DATA CONDITION

Abstract *Due to the huge number of financial transactions, it is almost impossible for humans to manually detect fraudulent transactions. In previous work, the datasets are not balanced and the models suffer from overfitting problems. In this paper, we tried to overcome the problems by tuning hyperparameters and balancing the dataset with a hybrid approach using under-sampling and over-sampling techniques. In this study, we have observed that these modifications are effective in getting better performance in comparison to the existing models. The MCC score is considered an important parameter in binary classification since it ensures the correct prediction of the majority of positive data instances and negative data instances. So, we emphasize on MCC score and our method achieved an MCC score of 97.09%, which is far more (16 % approx.) than other state-of-the-art methods. In terms of other performance metrics, the result of our proposed model has also improved significantly.*

Keywords credit card, fraud detection, deep learning, fraud transactions

Citation Computer Science 25(2) 2024: 253–276

1. Introduction

In the banking system, a credit card is a plastic card issued by banks or any other financial institution. It enables the users to borrow funds from the concerned institutions for availing of any kind of goods and services that the user wants. Credit cards are issued with a condition that the cardholder must pay back the original borrowed amount along with some additional interest amount which is decided by the banks or financial institutions. In the modern world of cashless transactions, credit cards play a very important role.

The total number of credit card users in India in 2019 touched around 52 million and it goes on increasing rapidly day by day. A credit card is a good option for the users for cashless pay, along with a good source for respective banks revenue generation. But with all the benefits and perks credit cards is very susceptible to fraud. Credit card frauds are easier to ensue in a short period and can cause excessive loss to both the cardholder and the bank. Fraudsters always focus on camouflaging fraudulent transactions as legitimate, so it becomes challenging to detect and stop them. The report published by the National Crime Records Bureau (NCRB), mentioned cases of online financial fraud using credit or debit cards have abruptly increased by over 225% in India during the pandemic period of 2020.

A credit card fraud is a fraudulent money transaction that is unauthorized and is carried out without the knowledge of the legal cardholder. These sorts of frauds can be conducted in various ways, it can be the thief stealing the card physically or stealing the card details or information via any means such as cyber-attack, etc. Although there is no foolproof technique to stop credit card fraud, but with the help of machine learning this issue can be addressed and has shown its relevance in the past. The main objective of any credit card fraud detection scheme is to identify or detect any sort of suspicious acts or events in the transactions and report them to the financial institution.

In literature, many credit card fraud detection techniques have been proposed and they claim very good performance. Among the most successful methods, many of them use machine learning models. The paper [22] shows the utilization of artificial neural networks and self-organizing maps for this purpose. The described model utilizes the advanced combination of neural networks for data analysis by the set of facts obtained from the previous transactions. The self-organizing maps are used for pictorial representation of the status of various organizations. According to the authors, the concept of self-organizing maps will be extremely efficient for anomaly detection.

The paper [25] used the optimized light Gradient Boosting Machine (GBM) where Bayesian-based hyper-parameter optimization is utilized in combination with tuned light GBM. This model achieves good accuracy and an F1 score. A comparative analysis among different machine learning algorithms to select which performs the best in detecting credit card fraud is proposed in [20]. A random forest technique-based fraud detection method is introduced in [14]. The proposed system achieved an accuracy of 90%. Cost-sensitive modelling using a neural network strategy for credit card fraud detection is projected in [10]. The cost-sensitive method initially deals with the imbalanced data issue. The neural network architecture uses two hidden layers in between the input and output layers. To study various neural networks, [5] provides a general idea about ANN, deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) for credit card fraud detection.

The credit card fraud detection problem contains developing and training a model using historical credit card transaction records with the knowledge of fraud transactions. Then the trained model is used to detect a new fraudulent transaction. Our aim here is to spot maximum fraudulent transactions while minimizing the incorrect fraud classification. This paper aims to develop a credit card fraud detection model by the knowledge gathered from reviewing various other models, which can identify suspicious events in credit card transactions.

It is well an established fact that better proportional neurons with the appropriate number of hidden layers give better results. With a greater number of hidden layers, more features can be extracted but it will be helpful up to a certain limit, beyond that instead of meaningful extracting features overfitting of data arises with increased system complexity. Overfitting in neural networks can the addressed with data augmentation, simplifying neural networks, weight regularization, dropouts, and early stopping. Overfitting is one of the main reasons for errors like false positives under limited imbalanced data conditions. In imbalanced data conditions results in terms of accuracy may be good due to overfitting but Matthew's correlation coefficient (MCC) score will be poor.

To overcome this issue in our study we have an emphasis on MCC score rather than other metrics and we balance the dataset to get proper results. In our study, we have also tried to optimize the neural network model to regularization weights of the neurons. To achieve this we also introduced a dropout layer in our proposed model. To speed up model training and hyperparameter optimization, we have also used the predictive early-stopping approach.

This paper is organized as follows; Section 2 presents the background research and related work, Section 3 presents details of existing models and their limitations, Section 4 presents the dataset and its pre-processing, and Section 5 demonstrates the description and development of the proposed model. Section 6 shows the comparison of existing models with our proposed model and Section 7 ends the work along with the conclusion and future directions.

2. Background and related work

Credit card fraud is a growing issue in today's modern world where many transactions are going online. Thus, it is very important to develop a full-proof and robust system that can detect and even predict patterns of any sort of credit card fraud.

Credit card fraud can be categorized in the following ways:

a) *PoS Fraud*: It is called Point-of-sale (PoS) fraud. A small skimming device is implanted into normal PoS devices, and these skimming devices hack the user's data. These devices scan and store the card data while customers are doing the transaction.

b) *Phishing and Vishing*: This fraud is accomplished through a cyber network. These involve mimicking the official communication from the bank which acts as an inducement for the target user. In the subsequent stage, the user is asked to click on a link, and these links redirect to a fake website (with an original webpage appearance) and request the user for card details.

c) *Keystroke Logging*: This usually happens when the user clicks on a suspicious link, and he/she unknowingly installs malware on the system. The malware secretly monitors every activity done by the user and records every key pressed on the system by the user, ultimately stealing the card details.

d) *Application Fraud*: In this type of fraud, a fraudster impersonates as a genuine client by using stolen or faked documents to get a credit card. In this fraud, the fraudster is using a legal credit card using false papers.

e) *Theft or Loss of a Card*: It is the physical misplacing of a credit card or getting stolen by someone. In this situation, there is always a chance of fraudsters taking advantage of it and using it to do fraudulent transactions.

Some of the previous research works done regarding credit card fraud detection are described and summarization of each of these is also discussed in this section. Paper [26] gives us a concrete idea about credit card fraud, how data is exploited, and the methodologies available for the detection of fraud. In the paper [19], several machine-learning techniques were used for the purpose. The model proposed a hybrid of K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and decision trees for credit card fraud detection. In the paper decision tree, Extreme Learning Machines (ELM), KNN, Multilayer Perceptron (MLP), and SVM were compared with their proposed hybrid model. It is observed in [16] that an imbalanced classification of the data leads to misleading and inaccurate results. They proposed that logistic regression, decision tree algorithm, and Artificial Neural Networks (ANN) are the best algorithms based on different performance matrices. The authors used data balancing for the training purpose of the model. The model proposed by [23] shows random forest algorithm is the best to provide higher accuracy rates in detecting fraud instances. They proposed an ensembling learning approach for the purpose. Ensembling learning consists of Random Forest and neural networks.

In the paper published by Suresh Kumar and Asha RB [4], ANN, SVM, and KNN algorithms were implemented on credit card data. The ANN model performs the best among the three algorithms. The model uses 30 features and the ANN architecture uses 15 hidden layers. The activation function in the model is Rectified Linear Units (ReLU). An autoencoder-based unsupervised fraud detection system based on clustering has been proposed in [30]. The autoencoders used three hidden layers and

K-means for the purpose of clustering. The models perform well compared to the other proposed models developed using the European dataset. In [15], Kernel-based supervised hashing (KSH) for this purpose of detection has been used. KSH models are useful in solving problems with huge data quantity and dimension. Compared to previous models proposed regarding credit card fraud detection, this is relatively a new approach. In the paper [21], the authors proposed thirteen models based on ANN and logistic regression. Results show the performance of the ANN models is superior when compared with logistic regression models.

A multilayer perceptron neural network utilization for fraud detection is used by [18]. A comparative study among multi-layer perceptron (MLP), decision tree, and naïve Bayes algorithms has been done. The accuracy of the MLP on the test data was more compared to other algorithms. In [9], ANN with backpropagation was used for credit card fraud detection. The neural network has three hidden layers. The first hidden layer has 15 neurons. In the model, the hidden layers use ReLU activation functions and in the output layer, the sigmoid activation function was used. The model achieves good accuracy and F1 Score.

From the above studies made on various proposed deep learning models, it is very forthcoming that deep learning models often suffer from overfitting or underfitting. Overfitting occurs when the model performs fine in training data but gives poor results in the testing conditions. In underfitting, the model performance is poor in training data and testing data, in underfitting the model is not trained enough to generalize to other data. In [13, 29], information related to the overfitting and underfitting of a model is provided. The authors mentioned how and why the overfitting and underfitting of models occur during the training phase. The effects of overfitting and underfitting are explained and discussed the techniques available to stop the overfitting and underfitting of the model. A study on the dropout and how it helps in reducing the chance of overfitting the neural networks is described in [24].

3. Baseline models for credit card fraud detection and their limitations

There has been extensive research work on this topic done previously and numerous articles have been published. Among the state-of-the-art literature methods, the three latest works [3, 4, 9] are considered for extensive study and the development of a new model for credit card fraud detection.

3.1. Existing model 1

In the paper [4], the authors proposed the ANN architecture with 15 hidden layers for credit card fraud detection. They used an imbalanced dataset and achieved an accuracy of 97.2% whereas the other parameters like precision and recall are not satisfactory and are very low. On calculating the F1 score, it is found to be 78.59%. The architecture of the model proposed by the above-mentioned work is depicted in the following Figure 1. The number of neurons in each hidden layer is not mentioned in detail, only the number of hidden layers and their activation function is revealed.

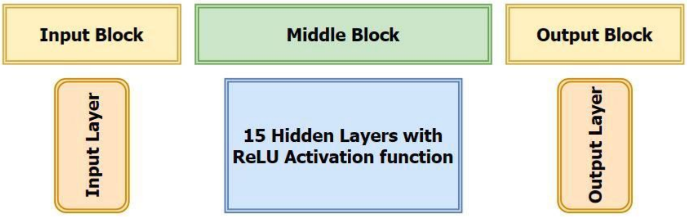


Figure 1. The architecture of baseline model-1

3.2. Existing model 2

The pictorial representation of the architecture mentioned in the paper [9] is shown below in Figure 2. The first hidden layer consists of 15 neurons and the activation function of all three hidden layers is ReLU. The diagram is constructed with the information provided by the authors in their paper. The model uses three hidden layers and achieves an accuracy of 97.2%. In this model, all the result parameters are close to ideal except the MCC score. The MCC value of the model is 81.46, which is very poor compared to the other metrics. Secondly, from the accuracy curve and the loss curve, it is clear that the curves are not smooth and tend to show some sort of overfitting which seems to be increasing with the number of epochs. Overfitting might tend to give inaccurate results when tested in a practical situation.

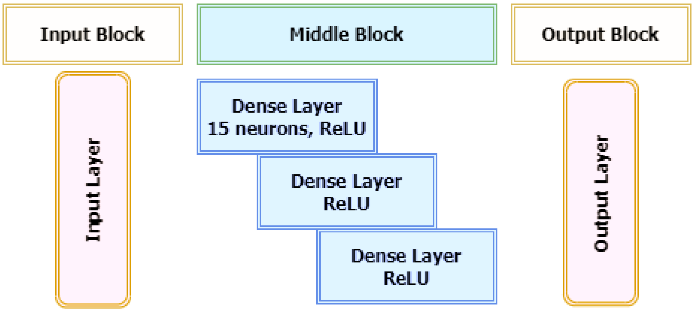


Figure 2. The architecture of baseline model-2

3.3. Existing model 3

In the paper [3], the authors proposed five hidden layers of ANN. The dataset is preprocessed and cleaned before being fed to train and test the model. The classifier has achieved a 95.3% accuracy. Furthermore, 95.2% of the fraud transactions are identified as shown by precision, and 95.55% times model is able to detect fraud transactions correctly as shown by recall rate. From the testing data, the model is also able to predict 95.5% of fraud transactions correctly. Although the model performance is good with all results reaching above the mark of 95%, it is comparatively low as compared to existing model-2. On the other hand, the MCC score is improved for this model in comparison to others. Secondly, the dataset used for validating the model is again used for the final performance evaluation of the model. In the testing phase, the element of uncertainty in data is missing, as the model has already seen the dataset during validation. The model contains seven dense layers which are linked consecutively and consist of 1024, 512, 256, 128, 64, and 32 neurons, out of which, five are hidden dense layers. The architecture of the hidden dense layers is shown in Figure 3.

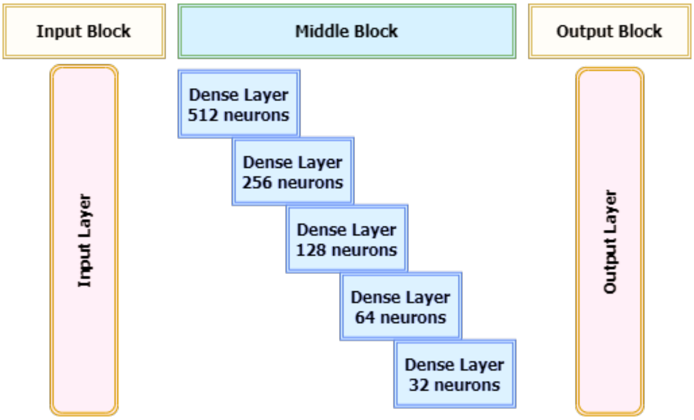


Figure 3. The architecture of baseline model-3

4. The dataset used for our analysis

The literature has employed a variety of datasets to evaluate and test credit card fraud detection methods. Typically, researchers use their own data. However, there are datasets that are freely accessible that might potentially be used. It is vital to note that having access to a reliable dataset is essential for enabling researchers to clearly compare their findings and demonstrate the superiority of one method over another. Finding datasets for financial research is challenging since the availability of such a dataset is still a problem in the field of credit card fraud detection. This is because the majority of banks do not share their data due to privacy issues and the sensitivity of client information. The data used for credit card fraud detection are of two types, real transaction data and Synthetic data. real transaction data are collected from real-world transactions whereas synthetic database is developed artificially rather than being gathered from actual situations. To develop a real-life system for credit card fraud detection, it is appropriate to develop a model on real-life data rather synthetic one. So, in this study, we have developed the model on real-life transaction data.

The dataset utilized for developing and validating the model is obtained from Kaggle [15], which contains information about various transactions made by European users in September 2013. The dataset has transactions that occurred for two days. The dataset contains 284807 transaction details, of which 492 fraud transactions and the remaining are legitimate. The dataset is highly imbalanced because the fraud class is much less compared to the legitimate class. The fraud is 0.172% of all the transaction records.

The dataset is Principal Component Analysis (PCA) transformed and contains only numerical data. For the sake of the confidentiality of the users, the original features and some of the background information of the data are not provided by Kaggle. The principal components obtained after PCA are features V1, V2, V3, ..., V27, and V28. Features such as 'Time' and 'Amount' are not PCA transformed. The dataset contains 31 feature columns and 284807 rows of credit card transaction details. In the column 'Class', fraud transactions are denoted as '1', and legitimate transactions are denoted as '0'.

4.1. Dataset preparation

As the dataset is highly imbalanced, directly using the data for training the model can lead to erroneous results. To avoid these possibilities, the dataset needs to be balanced before usage. When the distribution of classes in the dataset is uneven, data imbalance results. According to [1], the class imbalance may be a natural phenomenon or result from the challenging nature of data collection due to high costs, privacy issues, and labor demands. The dataset of credit card transactions is frequently unbalanced since there are relatively few fraudulent transactions compared to legitimate ones. A variety of techniques, including under-sampling [27] and over-sampling [7], are proposed to address this problem. The extremely unbalanced credit card datasets and the fact that each instance of the dataset contains important data (such as transactions held by the same cardholder) make these solutions difficult to implement [28]. The third data balancing approach is the hybrid of under-sampling and over-sampling techniques. In this analysis, we use the hybrid approach for data balancing. The method proposed in [17] is implemented to balance our dataset. The author used Random Under Sampling (RUS) to test numerous SMOTE variants, including the original variant, the borderline1 and borderline2 variants, SVM-SMOTE, SMOTEENN, and SMOTETomek. A hybrid balancing model was evaluated using the Balanced Bagging Ensemble, which is internally balanced using RUS and SMOTE. The outcomes demonstrated the scalability and superior performance of hybrid approaches.

From the dataset, 10% of the data containing both fraud and legitimate transactions were sorted and stored in a separate CSV file to test the model. The transaction details available in the testing dataset are then balanced to get 28126 fraud and 28481 legitimate transactions and a total of 56607 transactions are available for testing purposes.

The transaction in the remaining 90% is again balanced by above mentioned technique, resulting in 243650 fraud transactions and 255834 legitimate transactions, a total of 499484 transactions. This data is further divided into training and validation datasets. The dataset is divided into 80% for training and 20% for validation purposes.

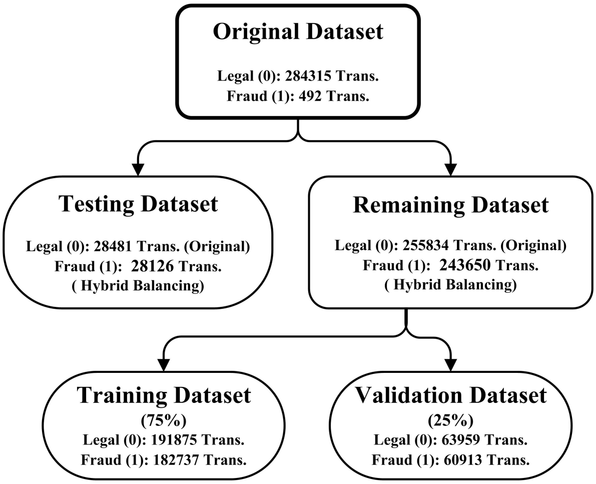


Figure 4. Representation of the dataset and its distribution into training, testing, and validation

5. Development of proposed model for credit card fraud detection

For the correct functioning of the neural network model, the number of hidden layers and associated neurons must be appropriate: neither more nor less. The development processes by which the number of hidden layers and associated neurons conform to the functional requirements of specific problems are complex and still, it is not addressed completely. To attain good results in neural networks a large number of hidden layers and neurons are required [11, 12]. We have studied and taken as a reference from a few research papers during the development of our proposed model. For finalizing the proposed models of credit card fraud detection, numerous deep learning models have been implemented. After extensive study associated with neural networks and simultaneously working on the drawbacks of the existing models, we came up with a final proposed model. After analyzing the three existing (baseline) models [3, 4, 9], it is observed that the existing (baseline) model-3 [20] is well described and properly explained. Therefore, necessary modification work is done on the existing (baseline) model-3 to build our proposed model. The other existing (baseline) models, namely 1 and 2 [4, 9] are used for comparison of results to show the improvement because these also used the same dataset.

The baseline model-3 consists of five hidden dense layers with one input and one output dense layer. Based on this, the investigative model-1 is developed which also consists of five hidden dense layers but the number of neurons in each layer is reduced. The results obtained from the investigative model-1 are analyzed and compared with existing model-1 and existing model-2, thereby encouraging further analysis because the results are not as expected.

To further improve the performance, the investigative model-2 is developed. This model uses six dense hidden layers with ReLU as the activation function. The investigative model-2 gives better results compared to our investigative model-1. However due to the increase in the number of layers and neurons, the computational time of the model is more, and the model takes more epochs to learn.

Further, we develop a model (Proposed Model) which consists of five dense hidden layers and one dropout layer. Now considering the results, it is seen that the proposed model is the best-performing model among all the designed and implemented models in terms of performance and computational time. The model also requires a lesser number of epochs to learn and performs better. To compare the existing and proposed models, Matthew's correlation coefficient (MCC) is one of the best parameters as it considers all the matrices (true positives, false negatives, true negatives, and false positives) available in the confusion matrix for evaluation. In the paper [8], a study regarding the reliability of MCC in binary classification has been done. The study suggests MCC is the most suitable parameter for binary classification since it ensures the correct prediction of the majority of positive data instances and the majority of negative data instances. Whereas, the F1 score merges precision and recall in a more interpretable way than MCC. For unknown or unquantifiable data conditions MCC is preferred over the F1 score as it is a more balanced calculation of classifiers, no matter which class is positive. So, we have given more emphasis on the MCC score over other performance evaluation matrices and decided on model performance on it. In Figure 5, we have represented the steps of the progressive development of our proposed models from the baseline model. It gives a clear picture of the design sequence of our analysis.

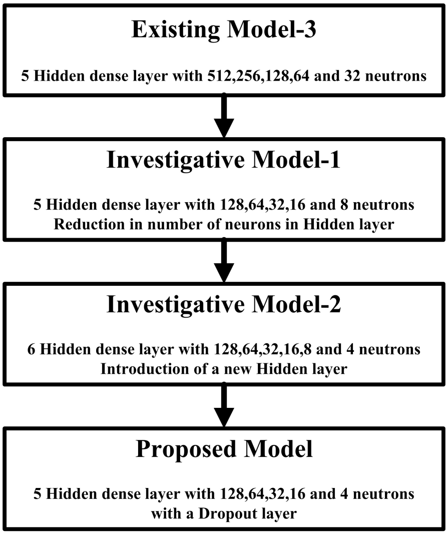

Existing Model-3
5 Hidden dense layer with 512,256,128,64 and 32 neutrons
Investigative Model-1
5 Hidden dense layer with 128,64,32,16 and 8 neutrons
Reduction in number of neurons in Hidden layer
Investigative Model-2
6 Hidden dense layer with 128,64,32,16,8 and 4 neutrons
Introduction of a new Hidden layer
Proposed Model
5 Hidden dense layer with 128,64,32,16 and 4 neutrons
with a Dropout layer

Figure 5. Progressive development of the proposed model

5.1. Description of investigative model-1 for credit card fraud detection

If a neural network model has too many neurons in the hidden layers results in overfitting, high variance, and increases the time it takes to train the network. This happens when the network has large information-handling capabilities but with limited training data, it is not possible to train the neurons in the hidden layers. In existing model-3, the number of neurons is very large as compared to the available data which results in inappropriate learning. So, in our investigative model-1, we have reduced the volume of neurons of each layer to address the issue. By selecting the proper activation function and other parameters, we can generate improved results as compared to the existing model-3.

The investigative model-1 consists of five hidden dense layers that are linked consecutively with 128, 64, 32, 16, and 8 neurons. The hidden layers contain the ReLU activation function, and the last layer is activated with the sigmoid activation function. The batch size of 700, 32 epochs, and the early stopping [13, 29] are used to circumvent the difficulties of overfitting and underfitting the model. Figure 6 shows the architecture of the investigative model-1.

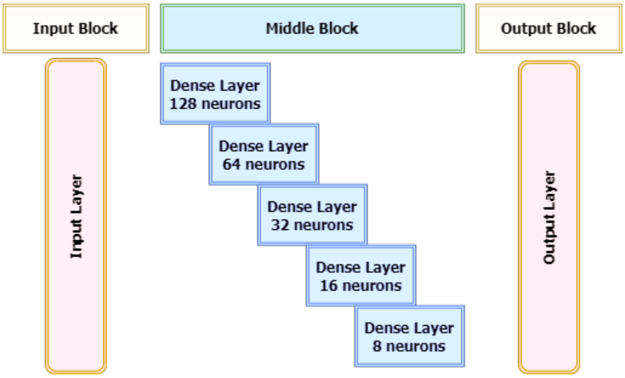


Figure 6. The architecture of Investigative model-1

The model when tested on the testing dataset gives satisfactory results (shown in Tables 1 and 2). Table 1 is the confusion matrix obtained for the investigative model-1, the model correctly identifies 28385 legitimate transactions which were genuinely legal and identifies 25256 transactions as fraudulent which were genuinely fraudulent. The model predicts only 96 fraudulent transactions which were actually legitimate and predicts 2870 transactions as legitimate but actually fraudulent.

Table 1

Confusion matrix of investigative model-1

		Predicted Label	
		Legitimate	Fraudulent
Actual Label	Legitimate	28385	96
	Fraudulent	2870	25256

Table 2 shows the result parameters of the investigative model-1, the model shows a precision of more than 99% and an F1 score of around 94% with an MCC value of 89.9%. but the accuracy of the model is slightly lower than the existing models. From the results, it is clear that our model is working fine and can differentiate between fraudulent and legal transactions.

Table 2

Result of investigative model-1

Accuracy	Precision	Recall	F1 Score	MCC
94.7603	99.6213	89.795	94.4537	89.947

The learning curves of the model shown in Figure 7 and Figure 8 also show that the model does not have any overfitting conditions. However the fluctuation of the validation curve shows that the model can be improved further, and the addition of an extra hidden layer may solve the issue raised.

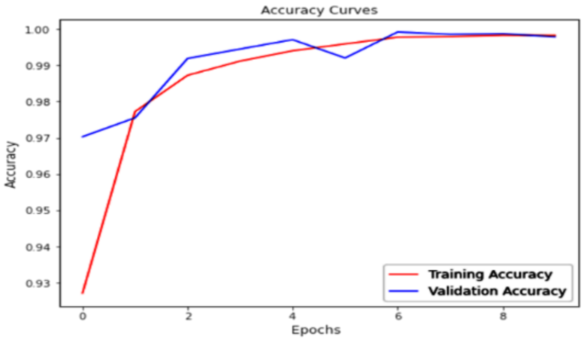


Figure 7. Accuracy curve of investigative model-1

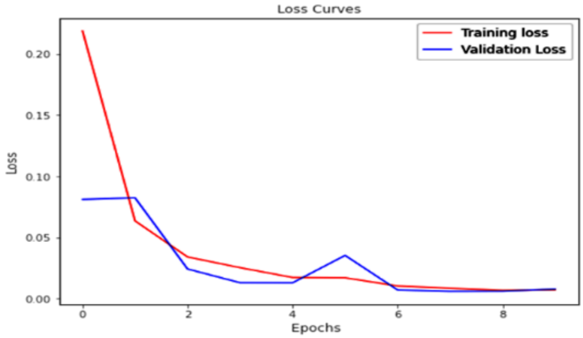


Figure 8. Loss curve of investigative model-1

5.2. Description of investigative model-2 for credit card fraud detection

In deep learning, better performance can be achieved if the entire problem is understood by the network without having overfitting and underfitting conditions [2, 12].

An increased number of hidden layers leads to an increase in the learning components that extract evidence from the previous activation to forward it to the next layer, which results in an improvement in performance and a reduction of bias. To handle big datasets and complex problems, a network with a large number of hidden layers normally shows high accuracy because the added layers provide more parameters to the model. And it allows the model to fit more complex functions. Inspired by this fact, we have increased the number of layers in investigative model-2 and checked the performance of the model with the same dataset (see Fig. 9).

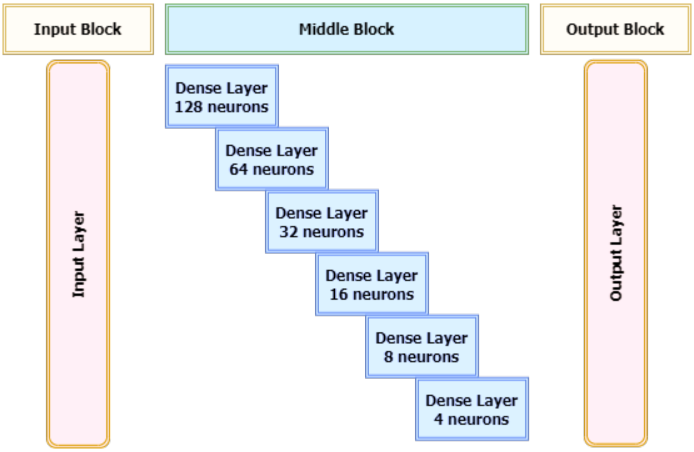


Figure 9. The architecture of investigative model-2

The investigative model-2 comprises six hidden dense layers that are linked consecutively with 128, 64, 32, 16, 8, and 4 neurons. The last layer is activated with a sigmoid activation function. With 32 epochs and 700 batch size, the early stopping technique [13, 29] is used to circumvent the difficulties of overfitting and underfitting the model. figure-9 shows the architecture of the investigative model-2.

The confusion matrix obtained for the investigative model-2 is shown in Table 3, the model correctly identifies 28378 legitimate transactions which were actually legal and identifies 26978 transactions as fraudulent which were actually fraudulent. The model predicts only 103 fraudulent transactions which were actually legitimate and predicts 1148 transactions as legitimate but actually fake.

Table 3
Confusion Matrix of Investigative Model-2

		Predicted Label	
		Legitimate	Fraudulent
Actual Label	Legitimate	28378	103
	Fraudulent	1148	26978

From Table 4 it is seen that the accuracy and F1 Score obtained is 97.7%, whereas the precision score is 99.6%. The results have improved when compared to the previous model. In the learning curves shown in Figure 10 and Figure 11, the number of epochs taken is around 17.5 which is early stopped. The curves are smoother thereby rejecting the possibility of overfitting. But as the number of hidden layers is increased the computational time of the model is increased.

Table 4

Result of Investigative Model-2

Accuracy	Precision	Recall	F1 Score	MCC
97.79	99.65	95.92	97.73	95.64

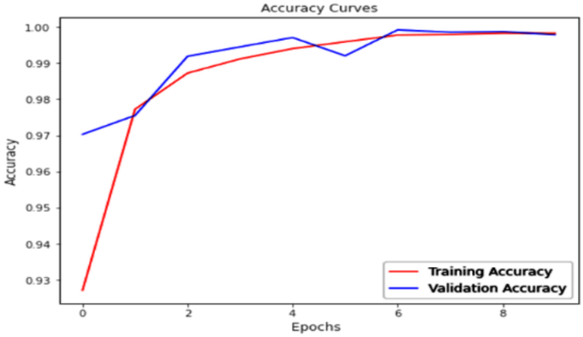


Figure 10. Accuracy curve of investigative model-2

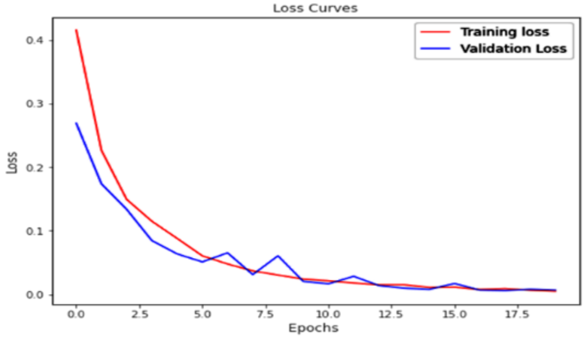


Figure 11. Loss curve of investigative model-2

5.3. Description of proposed model for credit card fraud detection

In many studies, it is observed that an increased number of layers provides a shortcut to improve the capabilities of the model with less amount of data. But it is also true that with the increased number of layers, the model may not extract meaningful features and system instead of learning the patterns, tries to memorize the information provided to it in training. One of the most important parameters of model designing is that the last dense layer must have the precise number of neurons and the proper activation function to get the best results. Each hidden neuron added will increase the number of weights, thus it is recommended to use the least number of hidden neurons that accomplish the task. Using more hidden neurons than required will add more complexity. Here in our proposed model, we have reduced the number of layers to five in comparison to investigative model-2 for better optimization. To further reduce the model complexity and improve performance, the introduction of the dropout layer is the solution [24]. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase. The dropout mechanism stops all neurons in a layer from synchronously optimizing their weights and results in gradual improvement in performance and reduction in loss [6]. Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons. If dropout is increased beyond a certain threshold results in improper model fitness. A higher dropout rate results in a higher variance to some of the layers and also degrades training process. To build a more capable network for better generalization and less likely to overfit the training data, we have used one dropout layer in our proposed model.

The architecture of the proposed model consists of five hidden dense layers that are connected sequentially with 128, 64, 32, 16, 4 neurons and one dropout layer. The last layer is activated with a sigmoid activation function. With 32 epochs and 700 batch size, and the early stopping [14], a dropout layer [21] is added to circumvent the difficulties of overfitting and underfitting the model. Figure 12 shows the architecture of the proposed model.

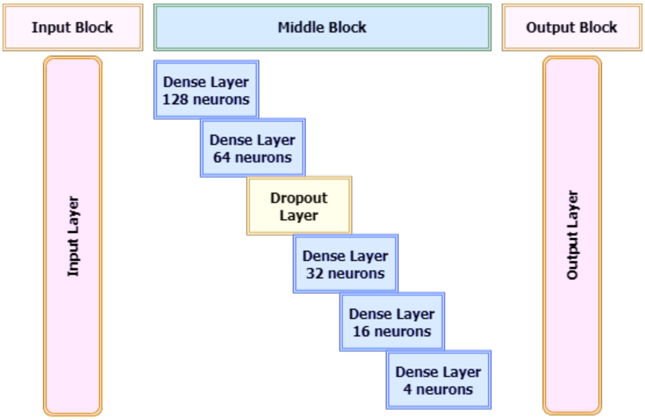


Figure 12. The architecture of the proposed model for credit card fraud detection

The detailed structure of the proposed model is shown in Table 5. The proposed feed forward neural network has the following architecture: the input to the model is text data with dimension 29. Here we have five dense layers connected sequentially with 3840, 8256, 2080, 528, and 68 learnable parameters in each layer respectively. The total number of learnable parameters in our model is 14777, which is sufficient to learn the patterns from the data. ReLU Activation function is used followed by all the dense layers for better generalization and to capture diverse patterns in the data, leading to better generalization in the presence of unseen examples. The final activation function in the output layer is 'sigmoid', indicating that the model is intended for binary classification of legitimate and fraudulent levels. After the second dense layer, we added one dropout layer with a rate of 0.25 that indicates 25% of neurons are randomly set to zero during training to prevent overfitting. The kernel initializer parameter is set to 'uniform' for all dense layers indicating that the weights of the neurons in those layers are initialized from a uniform distribution.

Table 5

The detailed structure of the proposed model for credit card fraud detection

Sl. No.	Layer (type)	Input Shape	Output Shape	Learnable Parameters
1	Input Layer	(None, 29)	(None, 29)	0
2	Dense Layer-1 ReLU Activation	(None, 29)	(None, 128)	3840
3	Dense Layer-2 ReLU Activation	(None, 128)	(None, 64)	8256
4	Dropout Layer Dropout rate: 0.25	(None, 64)	(None, 64)	0
5	Dense Layer-3 ReLU Activation	(None, 64)	(None, 32)	2080
6	Dense Layer-4 ReLU Activation	(None, 32)	(None, 16)	528
7	Dense Layer-5 ReLU Activation	(None, 16)	(None, 4)	68
8	Output Layer Sigmoid Activation	(None, 4)	(None, 1)	5

The confusion matrix obtained for the proposed model is shown in Table 6, the model correctly identifies 28229 legitimate transactions which were actually legal and identifies 27552 transactions as fraudulent which were actually fraudulent. The model predicts only 252 fraudulent transactions which are actually legitimate and predicts 574 transactions as legitimate that are fraudulent.

Table 6

Confusion matrix of proposed model

		Predicted Label	
		Legitimate	Fraudulent
Actual Label	Legitimate	28229	252
	Fraudulent	574	27552

The model shown in Figure 12 gives an accuracy of 98.5% and precision of 99.1% with a recall of 97.96% and F1 scores tending around 99%. The MCC score obtained is around 97.09% which is the highest among all models. The learning curves shown in Figure 13 and Figure 14, signify no overfitting. Although there is an initial spike in the learning curve other than this the learning curves get smoother with each epoch and finally early stopped at epoch 13. The model takes less time to get trained compared to the other models and also generates better results in all aspects.

Table 7
Result of the proposed model

Accuracy	Precision	Recall	F1 Score	MCC
98.54	99.09	97.96	98.52	97.09

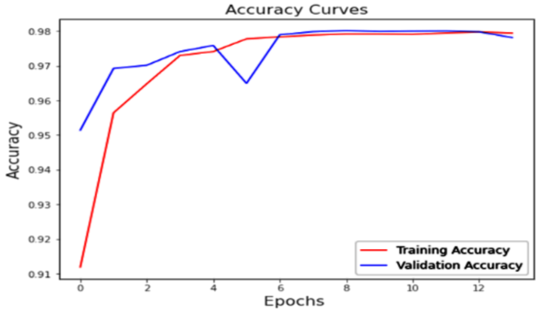


Figure 13. Accuracy curve of the proposed model

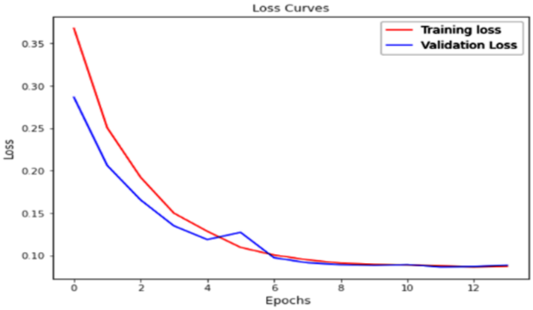


Figure 14. Loss curve of the proposed model

6. Comparison of existing models with our models

In this section, we have discussed the results obtained from our analysis, and Table 8 compares the results obtained from the investigative models and proposed model with existing models highlighting the limitations of existing models and improvements of our proposed model.

Table 8
Result comparison of different credit card fraud detection models

	Accuracy	Precision	Recall	F1 Score	MCC	
Baseline Models						Limitations
Model-1 [4]	97.23	81.15	76.19	78.59	80.26	• The use of a highly imbalanced dataset leads to high accuracy
Model-2 [9]	97.2	97.96	97.96	97.96	81.46	• No data preprocessing is done; directly using the imbalanced data leads to model overfitting (the model considers every transaction as legitimate). • The model is overfitted and can be observed from the learning curve. • Less amount of testing data
Model-3 [3]	95.3	95.2	95.55	95.5	82.47	• The dataset is divided into Training (70%) and Testing (30%), both are used for training and validating the model. The validation data is again used for evaluating the model performance
Factfinding Models						Refinements
Investigative Model-1	94.76	99.62	89.80	94.45	89.95	• In the proposed model, the dataset is pre-processed and balanced before training the model. • Early stopping during the training process and one dropout layer is introduced between hidden layers to stop overfitting or underfitting the model. • The number of neurons is optimized in the hidden layers for better performance (converges fast)
Investigative Model-2	97.79	99.61	95.92	97.73	95.64	
Proposed Model	98.54	99.09	97.96	98.52	97.09	

In this comparison, we have first identified the limitations of the existing models. The limitation of existing model-1 is in its training dataset. As the dataset consists of a very small number of fraud transactions, the model is not able to learn about fraud transaction patterns. Data preprocessing is not performed for existing model-2 and uses imbalanced data that leads to model overfitting. And the results obtained are on very less testing samples which leads to erroneous results. The shortcoming of the existing model-3 is the use of the same data for validation and testing the model. In testing, the model is not exposed to unknown data that gives unreliable results. To overcome these limitations, we have suggested a suitable model with adequate pre-processing and an optimized learning procedure. Before training the model, the dataset for the proposed model is pre-processed and balanced. To prevent the model from overfitting and underfitting, early stopping is implemented during the training phase, and one dropout layer is included in between hidden layers. For improved performance (rapid convergence), the number of neurons in the hidden layers is adjusted by different sets of investigative models and finally proposed an optimized model with a definite number of neurons in each dense layer. We have achieved significant improvements in model performance with limited trainable parameters.

The Matthews Correlation Coefficient (MCC) is a more dependable statistical rate that conveys a high score only if the estimate is accurate in all of the four confusion matrix classes and both the size of the positive class and negative class [8] are proportional. Figure 15 shows the pictorial representation of the results obtained for existing models, investigative models, and the proposed model.

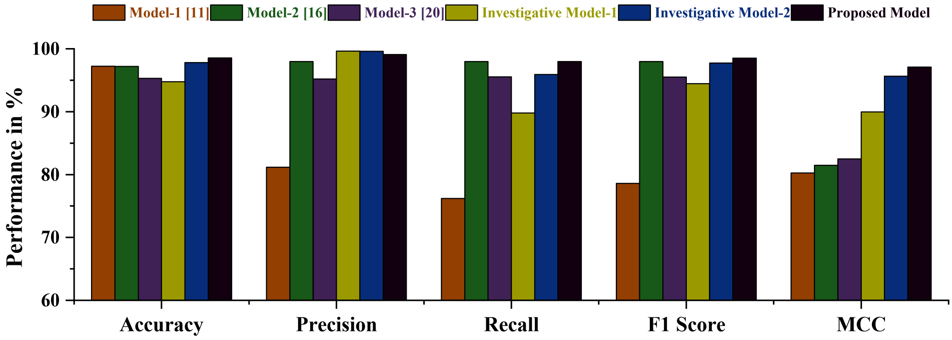


Figure 15. Performance analysis of various models

Comparing the MCC values, it is pragmatic that the proposed model is much better when compared with the existing models. We have achieved significant improvement of 16.83%, 15.63%, and 14.62% in MCC, compared to existing model-1, 2 and 3 respectively.

The accuracies of the existing models 1 and 2 are very close to the proposed model, this is due to the balanced datasets which are used to train the proposed model whereas, for the existing models, the dataset is highly imbalanced. From the results shown in Table 8, it is observed that the proposed model has a comparatively better precision value than the existing model-1 (improved 17.94%) and decent enhancement in precision score with respect to existing model-2 (improved 1.13%) and existing model-3 (improved 3.89%). This shows that our model is more capable of differentiating between false positives and true positives. Although achieving both high precision and high recall at the same time is difficult, but still our system is capable of maintaining the recall to a comparatively good value (slightly lesser than precision). The F1 score considers both precision and recall in its calculation. The F1 Score obtained for the proposed model is also upright and better than all existing models (19.93% for existing model-1, 0.56% for existing model-2, and 3.02% for existing model-3).

Due to a smaller number of fraud transactions compared to genuine transactions in real-world circumstances, most of the available datasets are highly imbalanced. So, data balancing is extremely important before training the model. The use of such imbalanced data can lead to a biased model towards genuine class, which consequent to wrong predictions. A clear picture of a biased model can be seen in existing model-1, although the accuracy of the model is more than 95.23%, but the other parameters are not satisfactory.

The 'Time' column in the dataset consists of seconds elapsed between subsequent transactions. On analyzing the effect of various features on the output, it is seen that the 'Time' information does not play any significant role in the performance of a model. Therefore, removing the 'Time' column from the dataset and using the other 29 features proposed in the model gives better results.

For effective learning maximum volume of data is essential to train the model. But to prevent the model from overfitting, an early stopping technique during the training is necessary. Comparing the learning curves of the existing model-2 [9] with the proposed model shows that the proposed model does not have an overfitting problem.

From the development of investigative model-2 to the proposed model, it is seen that the introduction of a dropout layer improves the results to a great extent. The proposed model uses one less hidden layer thereby reducing the computation complexity.

From the comparison of the proposed model with the three existing models, we can draw the following conclusion:

a) The accuracy of the proposed model is better than the existing models.

b) The precision score of the proposed models is improved to a great extent than the existing models.

c) The recall values and the F1 Scores of the proposed model are slightly better than the existing model-2, 3 and much more than the existing model-1.

d) The MCC score of the proposed model is very high related to all the existing models (16.83%, 15.63%, and 14.62%). The MCC value of the existing model-2 decreases to 80.26% whereas all other performance parameters of it are around 97%. This variation shows the model discrepancy in prediction.

e) We have accomplished our analysis on balanced data, but existing model results are on unbalanced data. So, our proposed model is more generalized to predict fraud efficiently.

7. Conclusion and future work

Credit card fraud is an important problem in the world, and modern machine learning techniques like neural networks are a potent way of detecting fraudulent transactions among a large number of transactions. From the literature, it is evident that neural networks like Deep Neural Networks (DNN) are more capable of establishing the association between input features and output even when there is no direct relationship between them. The development of our work started with several motivations: to develop a good and efficient model, to study different techniques used previously for this purpose, to rectify the limitations of the considered existing models, and to design a neural network to solve the problem of credit card fraud detection.

Whenever it comes to binary classification, the MCC score plays a vital role in determining the predicting capacity of the model. From the paper published by Davide Chicco and Giuseppe Jurman [8], it is evidenced that the MCC is a more reliable statistical parameter that produces a high score only if the model predicts effectively. Based on the MCC score, it is conclusive that our proposed model is much better compared to other existing models. Our proposed model achieves much more reliable and satisfactory results in terms of all parameters in comparison to other models.

Even after the finest efforts made in developing a robust credit card fraud detection model, there are still some limitations and shortcomings present. To train our model, the dataset is manually balanced, but in real-life circumstances, manual balancing is not feasible, so the use of other data balancing techniques like SMOTE and ADASYN can be implemented. Secondly, training the model on more available data will help in finding the drawbacks of the model. Finally, some traditional machine learning techniques like Decision Tree and Support Vector Machines (SVM) are also capable of giving good results, so developing a hyper-model consisting of neural networks and Decision Tree/SVM will help in developing a robust and efficient credit card fraud detection model. Though this study is concentrated on credit card fraud detection, the method developed in our study can help detect and protect the interest of any other plastic or virtual card holders.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest

The authors declare no competing interests.

Authors' contributions

All authors have contributed to this work

Funding

There is no external fund for this research.

Data availability statement

https://www.kaggle.com/datasets/mlg-lb/creditcardfraud

References

[1] Abd Elrahman S.M., Abraham A.: A review of class imbalance problem, *Journal of Network and Innovative Computing*, vol. 1(2013), pp. 332–340, 2013.

[2] Adil M., Ullah R., Noor S., Gohar N.: Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design, *Neural Computing and Applications*, pp. 1–9, 2022.

[3] Alkhatib K.I., Al-Aiad A.I., Almahmoud M.H., Elayan O.N.: Credit Card Fraud Detection Based on Deep Neural Network Approach. In: *2021 12th International Conference on Information and Communication Systems (ICICS)*, pp. 153–156, IEEE, 2021. doi: 10.1109/ICICS52457.2021.9464555.

[4] Asha R.B., Suresh Kumar K.: Credit Card Fraud Detection Using Artificial Neural Network, *Global Transitions Proceedings*, vol. 2(1), pp. 35–41, 2021. doi: 10.1016/j.gltp.2021.01.006.

[5] Aslam S., Herodotou H., Ayub N., Mohsin S.M.: Deep learning based techniques to enhance the performance of microgrids: a review. In: *2019 International Conference on Frontiers of Information Technology (FIT)*, pp. 1160–1165, IEEE, 2019. doi: 10.1109/fit47737.2019.00031.

[6] Brownlee J.: Dropout Regularization in Deep Learning Models with Keras, *Machine Learning Mastery*, vol. 20, 2016.

[7] Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P.: SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. doi: 10.1613/jair.953.

[8] Chicco D., Tötsch N., Jurman G.: The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation, *BioData Mining*, vol. 14(1), 13, 2021. doi: 10.1186/s13040-021-00244-z.

[9] Dubey S.C., Mundhe K.S., Kadam A.A.: Credit card fraud detection using artificial neural network and backpropagation. In: *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 268–273, IEEE, 2020. doi: 10.1109/iciccs48265.2020.9120957.

[10] Ghobadi F., Rohani M.: Cost sensitive modeling of credit card fraud using neural network strategy. In: *2016 2nd international conference of signal processing and intelligent systems (ICSPIS)*, pp. 1–5, IEEE, 2016. doi: 10.1109/icspis.2016.7869880.

[11] Gupta T.K., Raza K.: Optimizing deep feedforward neural network architecture: A tabu search based approach, *Neural Processing Letters*, vol. 51, pp. 2855–2870, 2020. doi: 10.1007/s11063-020-10234-7.

[12] Huang G.B.: Learning capability and storage capacity of two-hidden-layer feedforward networks, *IEEE Transactions on Neural Networks*, vol. 14(2), pp. 274–281, 2003. doi: 10.1109/tnn.2003.809401.

[13] Jabbar H., Khan R.Z.: Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study), *Computer Science, Communication and Instrumentation Devices*, vol. 70, 2015. doi: 10.3850/978-981-09-5247-1_017.

[14] Kumar M.S., Soundarya V., Kavitha S., Keerthika E.S., Aswini E.: Credit card fraud detection using random forest algorithm. In: *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*, pp. 149–153, IEEE, 2019. doi: 10.1109/iccct2.2019.8824930.

[15] Li Z., Liu G., Wang S., Xuan S., Jiang C.: Credit Card Fraud Detection via Kernel-Based Supervised Hashing. In: *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 1249–1254, IEEE, 2018. doi: 10.1109/smartworld.2018.00217.

[16] Makki S., Assaghir Z., Taher Y., Haque R., Hacid M.S., Zeineddine H.: An experimental study with imbalanced classification approaches for credit card fraud detection, *IEEE Access*, vol. 7, pp. 93010–93022, 2019. doi: 10.1109/access.2019.2927266.

[17] Mohammed R.A., Wong K.W., Shiratuddin M.F., Wang X.: Scalable Machine Learning Techniques for Highly Imbalanced Credit Card Fraud Detection: A Comparative Study. In: X. Geng, B.H. Kang (eds.), *PRICAI 2018: Trends in Artificial Intelligence. PRICAI 2018,* Lecture Notes in Computer Science, pp. 237–246, Springer, Cham, 2018. doi: 10.1007/978-3-319-97310-4_27.

[18] Mubarek A.M., AdalĬ E.: Multilayer perceptron neural network technique for fraud detection. In: *2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 383–387, IEEE, 2017. doi: 10.1109/ubmk.2017.8093417.

[19] Prusti D., Rath S.K.: Web service based credit card fraud detection by applying machine learning techniques. In: *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pp. 492–497, IEEE, 2019. doi: 10.1109/tencon.2019.8929372.

[20] Sadgali I., Nawal S., Benabbou F.: Fraud detection in credit card transaction using machine learning techniques. In: *2019 1st International Conference on Smart Systems and Data Science (ICSSD)*, pp. 1–4, IEEE, 2019. doi: 10.1109/icssd47982.2019.9002674.

[21] Sahin Y., Duman E.: Detecting credit card fraud by ANN and logistic regression. In: *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 315–319, IEEE, 2011. doi: 10.1109/inista.2011.5946108.

[22] Saraswathi E., Kulkarni P., Khalil M.N., Nigam S.C.: Credit card fraud prediction and detection using artificial neural network and self-organizing maps. In: *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1124–1128, IEEE, 2019. doi: 10.1109/iccmc.2019.8819758.

[23] Sohony I., Pratap R., Nambiar U.: Ensemble learning for credit card fraud detection. In: *CODS-COMAD '18: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pp. 289–294, 2018. doi: 10.1145/3152494.3156815.

[24] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *The Journal of Machine Learning Research*, vol. 15(56), pp. 1929–1958, 2014. http://jmlr.org/papers/v15/srivastava14a.html.

[25] Taha A.A., Malebary S.J.: An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine, *IEEE Access*, vol. 8, pp. 25579–25587, 2020. doi: 10.1109/access.2020.2971354.

[26] Xinwei Z., Yaoci H., Xu W., Qili W.: HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture, *Information Sciences*, vol. 557, pp. 302–316, 2021. doi: 10.1016/j.ins.2019.05.023.

[27] Yen S.J., Lee Y.S.: Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In: *Intelligent Control and Automation: International Conference on Intelligent Computing, ICIC 2006 Kunming, China, August 16–19, 2006*, pp. 731–740, Springer, 2006.

[28] Yeşilkanat A., Bayram B., Köroğlu B., Arslan S.: An adaptive approach on credit card fraud detection using transaction aggregation and word embeddings. In: I. Maglogiannis, L. Iliadis, E. Pimenidis (eds.), *Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology*, pp. 3–14, Springer, 2020. doi: 10.1007/978-3-030-49161-1_1.

[29] Ying X.: An overview of overfitting and its solutions, *Journal of Physics: Conference Series*, vol. 1168(2), 22022, 2019. doi: 10.1088/1742-6596/1168/2/022022.

[30] Zamini M., Montazer G.: Credit card fraud detection using autoencoder based clustering. In: *2018 9th International Symposium on Telecommunications (IST)*, pp. 486–491, IEEE, 2018. doi: 10.1109/istel.2018.8661129.

Affiliations

Nirupam Shome
Assam University, Department of Electronics and Communication Engineering, Silchar 788011, India, nirupam-shome@yahoo.com

Devran Dey Sarkar
Assam University, Department of Electronics and Communication Engineering, Silchar 788011, India, devransarkar@gmail.com

Richik Kashyap
Assam University, Department of Electronics and Communication Engineering, Silchar 788011, India, rknits2010@gmail.com

Rabul Hussain Laskar
National Institute of Technology, Department of Electronics and Communication Engineering, Silchar, 788010, India, rhlaskar@ece.nits.ac.in

Received: 08.12.2023
Revised: 06.02.2024
Accepted: 20.02.2024

ShivKishan Dubey
Narendra Kohli

CLUSTERING FOR CLARITY: IMPROVING WORD SENSE DISAMBIGUATION THROUGH MULTILEVEL ANALYSIS

Abstract *Existing Word Sense Disambiguation (WSD) techniques have limits in exploring word-context relationships since they only deal with the effective use of word embedding, lexical-based information via WordNet, or the precision of clustering algorithms. In order to overcome this limitation, the study suggests a unique hybrid methodology that makes use of context embedding based on center-embedding in order to capture semantic subtleties and utilizing with a two-level k-means clustering algorithm. Such generated context embedding, producing centroids that serve as representative points for semantic information inside clusters. Additionally, go with such captured cluster- centres in the nested levels of clustering process, locate groups of linked context words and categorize them according to their word meanings that effectively manage polysemy/homonymy as well as detect minute differences in meaning. Our proposed approach offers a substantial improvement over traditional WSD methods by harnessing the power of center-embedding in context representation, enhancing the precision of clustering and ultimately overcoming existing limitations in context-based sense disambiguation.*

Keywords embeddings, center embedding, multilevel clustering, word sense disambiguation (WSD), polysemy and homonymy

Citation Computer Science 25(2) 2024: 277–300

1. Introduction

Word Sense Disambiguation (WSD) is a fundamental task in Natural Language Processing (NLP) that attempts to determine the correct meaning of a given word given its context [19]. Related studies have led to the creation of algorithms/methodologies that use a range of resources like knowledge based that employ various relations in terms of is-a/part-of, and corpus-based that having with sense-tagged information. However, the materials needed for these methods must be created by hand by people, making them costly to acquire and maintain. Distributional techniques are best alternative to this, which distinguish words according to their meanings on the grounds that words that appear in similar instances will also have comparable meanings. Further, even while most VSMs are helpful, they all have the drawback of having just one vector for each word, which blatantly misses the mark when it comes to polysemy [8]. In accordance to this, multi-prototype VSM are with hybrid methodologies introduced with WSD tasks.

Polysemy is the phenomenon in which a singular term has multiple meanings that are interrelated. For instance, *"bank"* can refer to both a *financial institution* and a *riverbank*. Disambiguating between these various meanings is essential for accurate language comprehension and subsequent NLP applications. Homonymy, on the other hand, occurs when words with diverse meanings share the same form. For example, *"bat"* can refer to both a *winged mammal* and *an item of sporting equipment*. Correctly resolving homonyms is essential for preventing misinterpretation and ensuring precise semantic comprehension. Following Figure 1, clearly described about density of clusters for word bank which reflects about its close connections in itself.

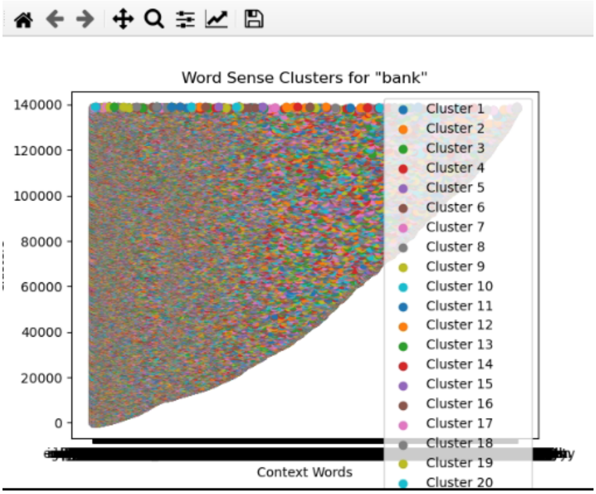


Figure 1. Cluster representation observed for ambiguous word *"bank"*. This represents word *"bank"* is how much overlapped and closed clusters in corpus due to its share contexts/senses

This figure clearly indicated that high degree of polysemy and the challenge of accurately disambiguating the word is reflected due to numerous senses or contexts associated with this ambiguous word *"bank"* and Close points indicated that the contexts associated with the word are similar or have overlapping characteristics due to the presence of related senses or the contextual similarities in the corpus.

Therefore, opted nested clustering [10] can provide a hierarchical structure to better organize and differentiate between different grouped contexts/senses, such contexts that represented as sentences even sometime complex sentences [7] which are hardly captured through traditional embeddings therefore to apply center embedding with context of left and right are very helpful to treat such grouped contexts. In this article, we propose with center-embedding based context embedding and utilize with nested level clustering (up to level 2) to address the challenges presented by polysemy and/or homonymy in WSD. Utilizing multilevel clustering permits us to generate clusters and sub-clusters that reflect the nuanced semantic relationships between words.

1.1. Provide the motivation

For Natural Language Processing (NLP) activities to advance, it is critical to address the shortcomings of conventional techniques to Word Sense Disambiguation (WSD). For precise language interpretation, information retrieval, machine translation, sentiment analysis, and other applications, NLP systems significantly rely on exact word sense disambiguation. However, these systems' efficiency and performance are hampered by the shortcomings of conventional techniques. Let's look at these restrictions in a Table 1 with some illustrations.

Table 1

Summerize view of limitations of traditional approaches

Limitations	Examples
Difficulty in capturing fine-grained distinctions among word senses	Consider the word *"run"*, which can have various senses such as *"to jog", "to manage", or "to operate"*. Traditional approaches often struggle to distinguish between such semantic differences, leading to incorrect interpretations [16]
Inability to handle high polysemy and homonymy	Traditional methods often fail to disambiguate words with multiple senses or different meanings. [20] e.g., the word *"crane"* can refer to *a bird* or *a large lifting machine*

To address these limitations, our proposed approach incorporates multilevel clustering which resolve such limitations (specially to capture polysemous, homonyms) in the proposed model of WSD:

Semantic Variations Captured: We find and group words that have similar semantic properties by using clustering methods to context word embeddings. This enables us to more properly capture the differences in word meanings within a certain context.

Handling Homonymy and Polysemy: Multilevel clustering makes it easier to recognize distinctive clusters and sub-clusters, allowing for a greater ability to distinguish between various meanings of polysemous and homonymous terms. By putting words in groups according to how they share a sense, it is easier to distinguish between terms like “bank” (financial organization vs. river bank) and ”bat” (flying animal vs. sports equipment).

Granularity: A more finely-grained representation of word senses is offered by the multilevel clustering technique. It enables the development of sub-clusters that can capture fine-grained semantic details inside a sense cluster. This granularity increases the accuracy of the disambiguation process and raises the overall effectiveness of NLP systems.

1.2. Our major contribution

The following are key objectives of our findings as:

- To develop a hybrid approach that makes use of multilevel clustering methods to increase the precision and level of word meaning disambiguation. This method will make it possible to interpret word meanings in various circumstances more precisely.
- We further address the difficulties presented by words having numerous meanings or senses (such contexts which represented in corpus with sentences even represent through complex sentences). Therefore, effectively capture such contexts through applying center-embedding based context embedding in multilevel clustering environment, and allowing to precisely comprehend and analyze their intended sense.
- To carry out in-depth analyses with detection of words having polysemy/homonyms and comparisons with current WSD techniques in order to gauge the performance and efficacy of our nested level clustering strategy.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work in the field of WSD. Section 3 presents the methodology and describes the multilevel center embedding approach in detail. Section 4 presents the experimental setup and the evaluation metrics used. Section 5 discusses the results and compares our approach with existing methods. Finally, Section 6 concludes the paper and discusses future directions for research in the field of enhanced word sense disambiguation.

2. Related previous works

Previous work in the area of word sense disambiguation (WSD) has looked into a number of strategies to increase the efficacy and accuracy of word sense disambiguation. Traditional approaches (in terms of graph-based, knowledge based and clustering based) have made a substantial contribution, but they frequently run into problems when dealing with polysemy and homonymy problems. Numerous research that are pertinent to our suggestion of adopting multilevel clustering for WSD have been done.

2.1. Knowledge based WSD approach

These approaches leverage the structured information within these resources to associate words with their appropriate sense and are very effective due to following reasons as rich semantic information,contextual disambiguation, transitivity of relations and no need for Sense-Tagged corpora. In terms of due to lack of contextual information and in cases of high polysemy or homonymy, where multiple senses of a word are plausible in a given context, knowledge-based methods may struggle to make precise disambiguation decisions.

In 2019, authors utilized genetic algorithms over wordnet and treated WSD task in Indo-aryan knowledge source specially for Gujrati language [21]. Later on, group of authors worked over SCSMM based approach [2] which combines sementic similarity, heuristics and document context to disambiguate terms.

2.2. Graph based WSD approach

Due to their ability to represent word meanings and context relationships in a structured manner. These approaches leverage the semantic relationships between words encoded in lexical knowledge bases like WordNet or other large-scale graphs built from text corpora. Some key reasons as word sense induction where senses are automatically induced from the graph's structure without relying on predefined sense inventories, easily represented relations of words and it's context in the form of nodes and edges that enables the model to consider the relatedness of different word senses and utilize this information to disambiguate the target word. Integration of external knowledge sources like WordNet, BabelNet, or domain-specific lexical resources that enhances the model's ability to access a rich pool of semantic knowledge and make informed disambiguation decisions.

As per above manners, group of authors utilizes random walks over large lexical knowledge base phenomenon using this graph based traditions [1] and got effective results during disambiguation task. To process over documents where authors [8] designed distributed graph based algorithms that cluster documents into same context and using heuristics vertex-centric algorithm inspired by metaphor of water cycle over WSD. In latest, graph based approach that proposed by authors [17] in which phenomenon utilized BFS to treated similarity and perform then WSD task very effectively.

Despite of effectiveness of graph based approach, words with multiple senses or ambiguous meanings can lead to densely connected graphs in traditional graph-based approaches, making sense disambiguation challenging. Another facts that can also make these approach in trouble when combines local context within clusters and global context between clusters which helps in disambiguating senses effectively therefore clustering approaches have good lead over this methods.

2.3. Clustering based WSD methods

A numerous cluster-based approaches have shown promising performance in WSD tasks that leveraged the inherent similarity and relatedness between instances to group them into clusters based on shared characteristics or context.

Following reasons explored why cluster-based approaches are effective in WSD.

- **Capture sensation Variation**: By grouping instances that have comparable contextual properties, cluster-based techniques may efficiently capture sensation variation [6].

- **Utilise Contextual Information**: To establish the sense of a target word by considering its surroundings. These methods can find patterns and relationships that help with disambiguation by examining the contextual parallels and divergences between occurrences.

- **Handle Polysemy and Homonymy**: In order to successfully handle these difficulties, cluster-based techniques divide instances into many clusters, each of which stands for a particular word meaning. As a result, more precise sense assignments may be made for terms having many meanings or ambiguous definitions.

- **Find Word Sense Relationships**: This approach can reveal hidden semantic connections between word senses. Hierarchies, similarities, and differences amongst senses may be found by looking at the connections between clusters and their sub-clusters. The process of disambiguation may be improved with the use of this information, which can also shed light on the semantic makeup of the intended term.

2.3.1. (Partially) supervised clustering

According to above effective facts, author proposed a classification based method which can utilized natural partitioning over mixed data (labeled and unlabeled) by maximizing stability criteria [15].

Method outperformed in order identification manner with semi-supervised k-means as base classifier. They investigated the stability criteria, which evaluates the degree of agreement between the sampled mixed data and the entire mixed data classification results. Following the assessment of the number of clusters, the ELP method was used to divide the mixed data into groups based on the estimated number of clusters, with each cluster being made up of comparable samples from the mixed data.

Later on, by investigated use of unlabeled data in semi supervised way through bootstrapping algorithm [10]. Another variations which associated with linear dimensionality reduction [4], where more separated clusters specially non-linear contexts are explored by group of authors through PMI based network clustering approach.

Utilizes a Semi-Autoencoder (SeAE) in the representation layer along with pair-wise constraint matrix based on PMI for accurate learning of distinctive features.

2.3.2. Unsupervised clustering

In 2004, group of authors [19] has designed unsupervised based clustering method which utilized context words surrounding noun and find relative candidates based on co-occurrence frequency which captured polysemy in noun effectively. They performed result analysis over WordNet. As getting effective performance over lexical resources further, where authors [3] overcome the sparseness of WordNet relations. Additionally they collect results for coarse-grained in English all-words task and fine-grained sub-task.

With the help of SVM classifiers over simple domain adaptation techniques, authors have given another aspect in such traditions where chosen senses utilizes for constructed clusters instances automatically [20]. Effective results are gained upto 74.7% precision score. Another unsupervised cluster based approach was explored by authors [13] where they utilized features' vector built from wordnet to represent senses as applicable over star clustering algorithm. Authors captured best performance among all unsupervised systems in SemEval-2007 with 72% F1-score.

In the area of information retrieval, authors have proposed unsupervised method for WSD [6] that utilized spectral clustering and reorders initially retrieved documents list by boosting documents that are semantically similar.

2.3.3. Other clustering methods

With the help of multilevel annotations markers [12] over lexical(lemma), tags(lex), grammatical tags, semantic taxonomy and combinations of these tags author effectively performed WSD task over RNC (Russian net corpus).

A very effective kernel based method over WSD was explored by authors [11] where they perform in-depth analysis and discussion of different strategies for representing context of polysemous words. Additionally, they explored kernel based strategies of feature selection and domain adaptation.

In 2017, a group of authors proposed sense inventory based unsupervised method [18], that utilized existing word embedding via clustering of ego-networks of related words.They got claimed effective results over the sense-balanced TWSI dataset as with 72% recall for WSD task.

Later on at very latest, authors proposed clustering based method by utilizing MFS (most frequent sense) of word [16] and treated with learned distribution which are effectively scored for other languages also.

2.4. Gap findings

Based on the previous work reviewed, there are several gaps and opportunities for further research on the proposed idea of using multilevel clustering for Word Sense Disambiguation (WSD).

Here is a summarized list that can supported for how multilevel clustering can be effective in addressing them:

- The successful organization of senses into hierarchies by multilevel clustering enables more subtle and precise disambiguation. It can ensure fine-grained sense distinctions.
- A number of methods need labeled data, which may be expensive and time-consuming. Multilevel clustering can lessen the demand for labeled data and make WSD more accessible, especially when used with unsupervised learning techniques.
- When words have homonymy and polysemous nature in fact in a very dense manner as Figure 1, traditional approaches may have trouble separating them. Such complexity may be managed by multilevel clustering, which efficiently captures sense variations and similarities hopefully.
- Multilevel clustering can more accurately capture context-dependent sense changes, resulting in more accurate disambiguation findings, by contextually grouping word senses.

Further in Section 3, proposed methodology along with detailed mathematical formulations are explored with multilevel clustering techniques that can lead to significant advancements in the field of WSD.

3. Methodology

The detailed mathematical formulation presented here outlines with following subsections of block diagram of proposed framework, formal notations, definitions and mathematical formulations as involved in multilevel clustering based WSD.

3.1. Block diagram of proposed framework

According to Figure 2, where to implement hybrid methodology of word sense disambiguation using three major modules as first to develop **text_instances**, for representing such instances to opt SemCor corpus which consists of a data file(.xml) with sense repository and this repository built through WordNet 3.0. This corpus also contain sentences with population as 37,176 into 352 different documents as docs.

Second main module of **context-embedding** that is considered here due to different sentences like simple and complex/compound type sentences are presented here, therefore go to next step for constructing context-embedding which is based on center-embedding with left and right contextual information inspired by [7]. Such context-embedding pass into clustering process where embedding of ambiguous word along with its context are operated in first level K-mean clustering. Process of how to calculate such center-embedding is discussed in next subsection 3.4.1 where The context embedding E_c is generated by applying a pre-trained (Spacy)word embedding model to each context token c_i. It capture the semantic information of the tokens and represent them as d-dimensional vectors.

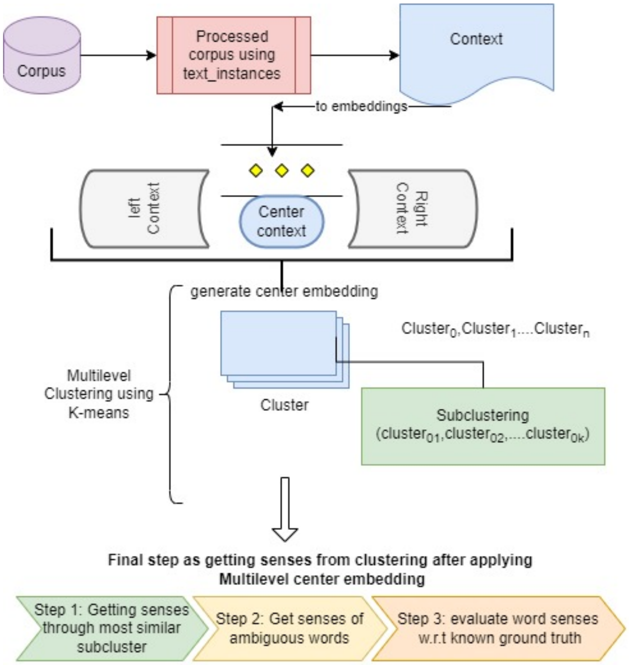

Corpus
Processed corpus using text_instances
Context
to embeddings
left Context
Center context
Right Context
generate center embedding
Multilevel Clustering using K-means
Cluster
$\text{Cluster}_0, \text{Cluster}_1 \ldots \text{Cluster}_n$
Subclustering ($\text{cluster}_{01}, \text{cluster}_{02}, \ldots \text{cluster}_{0k}$)
Final step as getting senses from clustering after applying Multilevel center embedding
Step 1: Getting senses through most similar subcluster
Step 2: Get senses of ambiguous words
Step 3: evaluate word senses w.r.t known ground truth

Figure 2. Block diagram for implementing Hybrid methodology of WSD task using Multilevel Clustering and center-embedding based context embedding. Figure consists with three major modules as instance collection from sense-directory, context embeddings, and nested level clustering

Now finally with significant module of **multilevel clustering** where the k-means algorithm minimizes the within-cluster sum of squares, which is equivalent to minimizing the squared Euclidean distances between data points and their respective cluster centers. Thus, by assigning definitions to the closest cluster centers, ensured that the definitions within each cluster are similar to each other and dissimilar from definitions in other clusters. At each level, the cluster centers are refined by computing the mean of the assigned definition vectors. This ensures that the new centers are representative of the definitions assigned to their respective clusters.

Applicability of proposed hybrid methodology, where nested level clustering with center-embedding is not only performed in levels but also in sub-levels, and finally go to capture most similar sub-clusters which are evaluated further with respect to known ground truth. This approach leverages the idea with center-embedding based context embedding into clustering procedure that the most common sense among similar definitions is likely to be the correct sense.

3.2. Optimal choice of number of level of clustering

As per proposed methodology, in which context-embedding is playing significant role to impose over clustering process. Proposed word sense disambiguation Algorithm-1 in further section that apply constraint between length of context embedding and number of clusters that needed as multilevel clustering. Therefore, we performed the task of getting frequent occurring of contexts up to clusters involvement as Figure 3, which clearly indicated here maximum number of contexts/senses (574361) are frequently lie at cluster 1, and then only with 2008 contexts are frequently observed when number of clusters is 2. Rest of contexts/senses which are very less frequent as 31, 6, 1 at number of clusters 3, 4 and 5 respectively. Therefore, with this observation, we decided to perform multilevel clustering upto level 2 which can performed over maximum possible contexts/senses through proposed hybrid methodology.

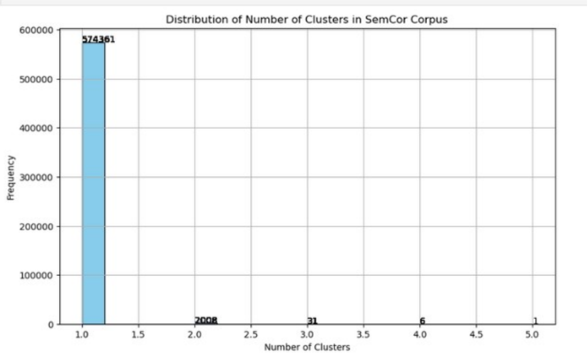


Figure 3. Reflecting Optimal choice of Nested-levels Clusters for the multilevel clustering sub-module of methodology

3.3. Formal notations

$$
\begin{cases}
C \Rightarrow c_1, c_2, ..., c_{N_c}, \text{ Set of context tokens \& } N_c \text{ is the number of context tokens.} \\
W \Rightarrow w_1, w_2, ..., w_{N_w}, \text{Set of word tokens, where } N_w \text{ is the number of word tokens.} \\
E_c \Rightarrow \text{Context embedding matrix,} \in \mathbb{R}^{N_c \times d}, \text{where } d \text{ is the dimensionality.} \\
E_w \Rightarrow \text{Word embedding matrix,} \in \mathbb{R}^{N_w \times d}, \\
k \Rightarrow \text{Number of clusters for context token embeddings.} \\
X_c \Rightarrow \text{Set of context embeddings as} \quad x_1, x_2, ..., x_{N_c}, \\
\text{where } x_i \text{ is the embedding for context token } c_i. \\
X_s \Rightarrow \text{Set of context embeddings in the most similar cluster as} \quad x_{1'}, x_{2'}, ..., x_M, \\
M \text{ is the number of embeddings in the most similar cluster.} \\
C_j \Rightarrow \text{Set of context embeddings assigned to cluster } j \text{ as} \quad x_1, x_2, ..., x_m, \\
m \text{ is the number of embeddings in cluster } j. \\
\mu_j \Rightarrow \text{Centroid of cluster } j \text{ as} \in \mathbb{R}^d, \\
\text{calculated as the mean of context embeddings in cluster } C_j.
\end{cases}
\tag{1}
$$

3.4. Mathematical formulation

After assigning notations, now these notations are utilizing in formulation of proposed idea. Our formulation described in terms of following steps as pre-processing, word embeddings, and context embeddings, calculating Cluster-Centroids, level-2 clustering, to assign mapping Between Word Tokens and Synsets, and finally to retrieve words in the Most Similar Cluster. Preprocess the context and w_a using appropriate techniques, such as tokenization, stemming, and stop-word removal.

3.4.1. Word and context embeddings

By taking these notations as context C consisting of context tokens c_i, where $i = 1, 2, ..., N_c$. Word embeddings E_w for each word token w_i, where $i = 1, 2, ..., N_w$ and to choose pre-trained word embedding model. Now the aim is to generate context embeddings that capture the contextual information by considering the neighboring words within the context [5].

Definition 1 (Context Window, Word Embeddings and calculation of Center embedding)**.** ***To extract Context Window:*** *For each context token c_i in the context C, extract the context window of size k, which includes the $(k-1)$ preceding and $(k-1)$ succeeding tokens around c_i. Let $CW(c_i)$ denote the context window extracted for the token c_i.*

Word Embeddings*: By representing each token in the context window $CW(c_i)$ using its corresponding word embedding. Let $E_w(c_i)$ denote the word embedding for the token c_i.*

Finally, go with ***Center Embedding Calculation*** *as for the context window $CW(c_i)$, take the average of the word embeddings within the window. Let $CE(c_i)$ represent the center embedding [7] for the token c_i.*

Proof. Let $CW(c_i) = c_{i-k}, ..., c_{i-1}, c_i\widehat{c_i}, c_{i+1}, ..., c_{i+k}$ be the context window extracted around the token c_i, where k is the size of the context window. For each token c_j in the context window $CW(c_i)$, we can represent its word embedding using $E_w(c_j)$. The center embedding $CE(c_i)$ can be calculated as the average of the word embeddings within the context window as

$$CE(c_i) = (1/(2k+1)) \cdot \Sigma_{c_j \in CW(c_i)}(E_w(c_j))$$

where Σ denotes summation and $(2k+1)$ represents the total number of tokens in the context window. By averaging the word embeddings within the context window, the center embedding captures the contextual information of the token c_i by considering its neighboring words. It allows us to represent the token in a vector space that encodes the semantic information within its local context. □

With this formulation, we can generate center embeddings for each context token in the given context. These center embeddings serve as the context embeddings, which can be used for further steps in the Word Sense Disambiguation task, such as clustering or similarity comparison.

3.4.2. Cluster centroid calculation

Definition 2. *Apply the k-means clustering algorithm to the context embeddings X_c with k clusters to obtain the cluster assignments and centroids at the first level.*

Proof. Given the context embeddings $X_c = x_1, x_2, ..., x_{N_c}$ and the number of clusters k, we can apply the k-means algorithm to X_c, which assigns each embedding to one of the k clusters and calculates the mean of the embeddings within each cluster. Let $C_1 = C_{1_1}, C_{1_2}, ..., C_{1_k}$ be the set of clusters at the first level, where each C_{1_j} represents a cluster with its assigned context embeddings. The centroid μ_{1_j} of cluster C_{1_j} is calculated as the mean of the context embeddings in C_{1_j} as following Equation (2).

$$\mu_{1_j} = (1/ \mid C_{1_j} \mid) \cdot \Sigma_{x_{i \in C_{1_j}}}(x_i) \tag{2}$$

where $\mid C_{1_j} \mid$ represents the cardinality of cluster C_{1_j}, i.e., the number of context embeddings assigned to cluster C_{1_j}. □

3.4.3. Second level clustering

Definition 3. *For each cluster C_{1_j} at the first level, apply the k-means clustering algorithm with a different number of clusters (e.g., $K_2 < k$) to the centroid embeddings μ_{1_j} obtained from the previous step. This results in the second level clusters C_{2_j} corresponding to each first level cluster C_{1_j}.*

Proof. Given the centroid embeddings $\mu_{1_j} = \mu_{1_{j_1}}, \mu_{1_{j_2}}, ..., \mu_{1_{j_{K_1}}}$ for the first level cluster C_{1_j}, we can apply the k-means algorithm to μ_{1_j} with K_2 clusters, where K_2 is the desired number of clusters at the second level. This assigns each centroid embedding to one of the K_2 clusters and calculates the mean of the embeddings within each cluster. Let $C_{2_j} = C_{2_{j_1}}, C_{2_{j_2}}, ..., C_{2_{j_{K_2}}}$ be the set of clusters at the second level corresponding to the first level cluster C_{1_j}. The centroid $\mu_{2_{j_k}}$ of cluster $C_{2_{j_k}}$ is calculated as the mean of the centroid embeddings in $C_{2_{j_k}}$ in Equation (3) as.

$$\mu_{2_{j_k}} = (1/ \mid C_{2_{j_k}} \mid) \cdot \Sigma_{\mu_{1_i} \in C_{2_{j_k}}}(\mu_{1_i}) \tag{3}$$

where $\mid C_{2_{j_k}} \mid$ represents the cardinality of cluster $C_{2_{j_k}}$, i.e., the number of centroid embeddings assigned to cluster $C_{2_{j_k}}$. □

3.4.4. Mapping between word tokens and synsets

As to perform lemmatization on each word token w_i to obtain its lemma form, denoted as $lemma(w_i)$. For each lemma form $lemma(w_i)$, find the corresponding synsets in WordNet, denoted as S_i. Establish then the mapping between word tokens and synsets, associating each word token w_i with its set of synsets S_i.

Definition 4. *To performs lemmatization on each word token w_i to obtain its lemma form, $lemma(w_i)$. It then finds the corresponding synsets in WordNet, denoted as S_i, which represents the possible senses of the word.*

Proof. Lemmatization is a linguistic process that transforms a word into its base or dictionary form. By applying lemmatization to each word token w_i, we obtain $lemma(w_i)$, which represents the base form of w_i. WordNet provides a lexical database that maps words to synsets, which are sets of synonymous words with a shared meaning. Thus, we can find the corresponding synsets S_i in WordNet for each $lemma(w_i)$, representing the possible senses of the word. □

3.4.5. Retrieve words in the most similar cluster

Definition 5. *For the ambiguous word w_a, calculate its embedding vector E_a using the pre-trained word embedding model. Calculate the dissimilarity between E_a and each centroid $\mu_{2_{j_k}}$ at the second level using cosine similarity. Select the most similar cluster C_{2_s}, which has the smallest dissimilarity, and retrieve the words in C_{2_s} as the disambiguated senses of the ambiguous word.*

Proof. Let E_a be the embedding vector for the ambiguous word w_a. Calculate the dis-similarity between E_a and each centroid $\mu_{2_{j_k}}$ using cosine similarity measure. Select the cluster C_{2_s} that minimizes the dis-similarity, i.e., $C_{2_s} = argmin(dis\text{-}similarity(E_a, \mu_{2_{j_k}}))$. Retrieve the words in C_{2_s} as the disambiguated senses of the ambiguous word. Let E_a be the embedding vector for the ambiguous word w_a, and let $\mu_{2_{j_k}}$ be the centroid embedding vector for cluster $C_{2_{j_k}}$ at the second level. The cosine similarity between E_a and $\mu_{2_{j_k}}$ can be calculated as following Equation (4).

$$similarity(E_a, \mu_{2_{j_k}}) = (E_a \cdot \mu_{2_{j_k}})/(\|E_a\| \cdot \left\|\mu_{2_{j_k}}\right\|) \tag{4}$$

where $\cdot$ denotes the dot product and $\|\|$ represents the Euclidean norm. The cosine similarity ranges between -1 and 1, with higher values indicating more similarity and lower values indicating more dis-similarity. However, we want to calculate dis-similarity, so we can define the dis-similarity as $1 - similarity(E_a, \mu_{2_{j_k}})$. This way, higher values indicate more dis-similarity. To find the most similar cluster, we iterate over all centroid embeddings $\mu_{2_{j_k}}$ and calculate the dissimilarity now and then to select the cluster C_{2_s} that minimizes the dissimilarity as following Equation (5).

$$C_{2_s} = argmin(1 - similarity(E_a, \mu_{2_{j_k}})) \tag{5}$$

Finally, we retrieve the words in cluster C_{2_s} as the disambiguated senses of the ambiguous word w_a.

□

3.5. Word sense disambiguation

As per above mathematical formulation, to implement with this following Algorithm 1 of the WSD, that is to determine the correct sense of the ambiguous word w_a within the given context C. It takes arguments as context, an ambiguous word, word embedding, cluster centroids, and a dissimilarity measure as input.

Algorithm 1 Word Sense Disambiguation (WSD) with nested level(2) clustering and center embedding calculation

```
function WSD($C$, $w_a$, $E_w$,$\mu_{1_j}$, $\mu_{2_{j_k}}$, dissimilarity measure)
                                                        ▷ To perform Pre-processing
    preprocess_context_tokens($C$)
    preprocess_ambiguous_word($w_a$)
                                                        ▷ Word and Context Embeddings
    $E_a \leftarrow$ calculate_embedding($w_a$, $E_w$)
                                                        ▷ First-level Clustering
    $min\_dissimilarity \leftarrow \infty$
    $most\_similar\_cluster \leftarrow$ None
    for each centroid $\mu_{1_j}$ in first-level cluster centroids do
        $dissimilarity \leftarrow$ calculate_dissimilarity($E_a$, $\mu_{1_j}$, dissimilarity measure)
        if $dissimilarity < min\_dissimilarity$ then
            $min\_dissimilarity \leftarrow dissimilarity$
            $most\_similar\_cluster \leftarrow$ cluster associated with $\mu_{1_j}$
                                    ▷ Retrieve Words in the Most Similar First-level Cluster
    $disambiguated\_senses\_level\_1 \leftarrow$ retrieve_words($most\_similar\_cluster$)
                                                        ▷ Second-level Clustering
    $min\_dissimilarity\_level\_2 \leftarrow \infty$
    $most\_similar\_subcluster \leftarrow$ None
    for each centroid $\mu_{2_{j_k}}$ in second-level cluster centroids do
        $dissimilarity\_level\_2 \leftarrow$ calculate_dissimilarity($E_a$, $\mu_{2_{j_k}}$, dissimilarity measure)
        if $dissimilarity\_level\_2 < min\_dissimilarity\_level\_2$ then
            $min\_dissimilarity\_level\_2 \leftarrow dissimilarity\_level\_2$
            $most\_similar\_subcluster \leftarrow$ subcluster associated with $\mu_{2_{j_k}}$
                    ▷ Return Words in the Most Similar Second-level/First level cluster(s)
    return $disambiguated\_senses\_level\_1$, $disambiguated\_senses\_level\_2$
function CALCULATE_EMBEDDING($w_a$, $E_w$)
    $target\_index \leftarrow$ index of $w_a$ in $tokens$
    $start\_index \leftarrow \max(0, target\_index - context\_window\_size)$
    $end\_index \leftarrow \min(\text{len}(tokens), target\_index + context\_window\_size + 1)$
    $context\_window \leftarrow tokens[start\_index : end\_index]$
    $word\_embeddings \leftarrow [E_w(c_j)$ for $c_j \in context\_window]$
    $center\_embedding \leftarrow \frac{1}{\text{len}(word\_embeddings)} \times \sum_{c_j \in context\_window} E_w(c_j)$
    return $center\_embedding$
```

It proceeds with initial tasks as *Tokenizes* and preprocesses the context C and ambiguous word. It generates the word embedding for ambiguous word, calculating the word and context embedding using the pre-trained (Spacy) word embeddings. Now furhter go with First-level Clustering where For each centroid μ_{1_j} in the first-level cluster centroids, calculate dissimilarity between word embedding and each centroid, and then to identify the cluster with the minimum dissimilarity as the most similar first-level cluster. Now to get the words in the most similar first-level cluster.

After go through with first level clustering go then second level clustering where for each centroids $\mu_{2_{j_k}}$ again to calculate dissimilarity between word embedding and each centroid $\mu_{2_{j_k}}$. Here we now identify the sub-cluster with the minimum dissimilarity as the most similar second-level sub-cluster. Finally return by selecting words in the most similar first level cluster and second-level sub-cluster, and the procedure returns the set of disambiguated senses of the ambiguous word w_a.

4. Experimental setup

Corpus: For Word Sense Disambiguation (WSD), the **Semcor** corpus is a frequently used benchmark dataset in computer linguistics and natural language processing. It is a piece of the Brown Corpus, a bigger corpus that includes literature from numerous genres. The Semcor corpus, which was created especially for WSD tasks, consists of English phrases that have WordNet lexical database word sense tags added to them [14]. It offers an important resource for analyzing and creating WSD models and algorithms. The Semcor corpus' main traits and qualities are as follows:

- Each word in the Semcor corpus has its associated WordNet sense explicitly tagged next to it.
- When several senses of a word are present in WordNet, the Semcor corpus concentrates on adding sense annotations to potentially ambiguous terms. The annotations try to clarify each uncertain word's appropriate meaning in the context.
- The Semcor corpus's annotations are in line with the WordNet sense inventory, which makes it easier to create and test WSD algorithms that rely on WordNet for sense definitions and distinctions.

Pre-trained embedding: A library of **spaCy** provide a valuable utility in natural language processing (NLP) tasks, including Word Sense Disambiguation (WSD). Here's a brief overview of the utility of pre-trained embeddings from spaCy:

- It can leverage this semantic information to better understand the meaning of words and their potential senses within a given context.
- The reduced dimensionality makes it computationally efficient to process and compare word embeddings, which is crucial in tasks like WSD where many word senses need to be considered.

As getting the source information from authors [19], where they give summarize three benchmark datasets as Semcor, Senseval, and Korena WN where maximum number of polysemous words are presented in Semcor with large number of instances population which can be helpful to identify senses. As with such data availability, we have opted to processed our proposed work in Semcor Corpus.

5. Results and discussion

In this article, we have first calculated performance score in terms of precision, recall and F1-score on SemCor corpus as a success indication. As a result, in our effort using the this corpus, we picked the target terms in unsupervised manner using multilevel clustering. For this reason, we have chosen a word as a target term only if its semantic differences are distinct in corpus. This dataset, in our opinion, provides more realistic accuracy results as compared to existing state-of-arts. The following findings in terms of various results are shown here with regard under sub-clusters up-to level-2 as the Algorithm 1.

5.1. Performance of proposed method

According to implement proposed WSD task using multilevel clustering over chosen corpus where (around 170000 sentences are correctly processed) to get overall F1-score as **88.36%** as Figure 4. Due to that, most of text (in green colored bars shown correctly captured while white color-strips reflected as not-captured senses) in corpus treated successfully under level-1/2 sub-clustering as proposed in Algorithm 1. Some of samples that are also getting predicted senses with individual performance score is represented below as Figure 5 where with respect to use a context, ambiguous word and along with known senses getting after executing Algorithm 1.

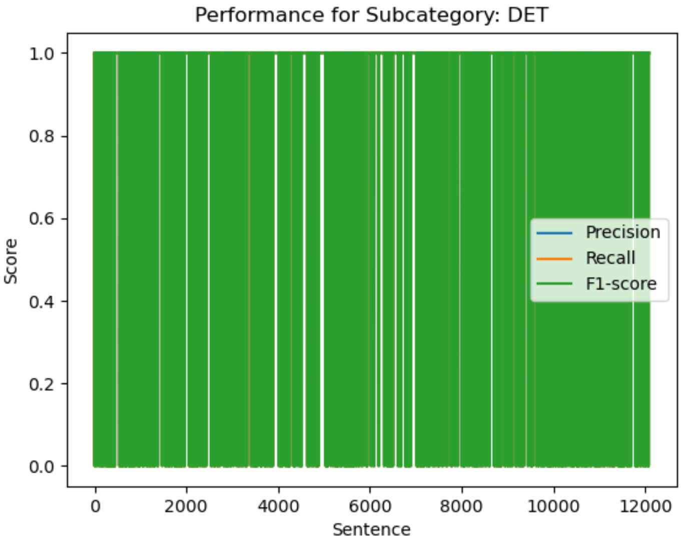
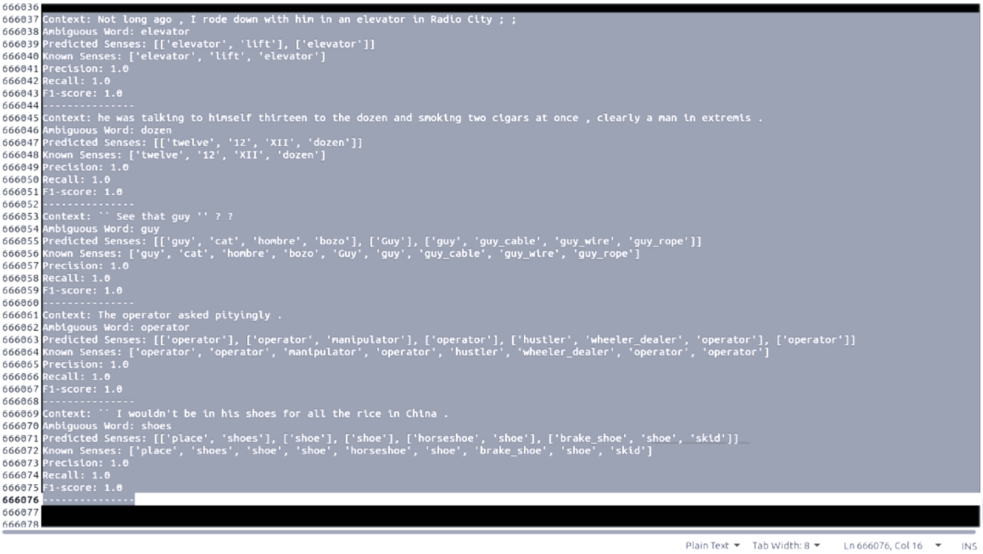


Figure 4. Visualization of Performance score of Proposed Hybrid methodology for WSD using multilevel clustering and center embedding based context embedding over sub-category sentences (with population of 12000) in SemCor Corpus



```
Context: Not long ago , I rode down with him in an elevator in Radio City ; ;
Ambiguous Word: elevator
Predicted Senses: [['elevator', 'lift'], ['elevator']]
Known Senses: ['elevator', 'lift', 'elevator']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: he was talking to himself thirteen to the dozen and smoking two cigars at once , clearly a man in extremis .
Ambiguous Word: dozen
Predicted Senses: [['twelve', '12', 'XII', 'dozen']]
Known Senses: ['twelve', '12', 'XII', 'dozen']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: `` See that guy '' ? ?
Ambiguous Word: guy
Predicted Senses: [['guy', 'cat', 'hombre', 'bozo'], ['Guy'], ['guy', 'guy_cable', 'guy_wire', 'guy_rope']]
Known Senses: ['guy', 'cat', 'hombre', 'bozo', 'Guy', 'guy', 'guy_cable', 'guy_wire', 'guy_rope']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: The operator asked pityingly .
Ambiguous Word: operator
Predicted Senses: [['operator'], ['operator', 'manipulator'], ['operator'], ['hustler', 'wheeler_dealer', 'operator'], ['operator']]
Known Senses: ['operator', 'operator', 'manipulator', 'operator', 'hustler', 'wheeler_dealer', 'operator', 'operator']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: `` I wouldn't be in his shoes for all the rice in China .
Ambiguous Word: shoes
Predicted Senses: [['place', 'shoes'], ['shoe'], ['shoe'], ['horseshoe', 'shoe'], ['brake_shoe', 'shoe', 'skid']]
Known Senses: ['place', 'shoes', 'shoe', 'shoe', 'horseshoe', 'shoe', 'brake_shoe', 'shoe', 'skid']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
```

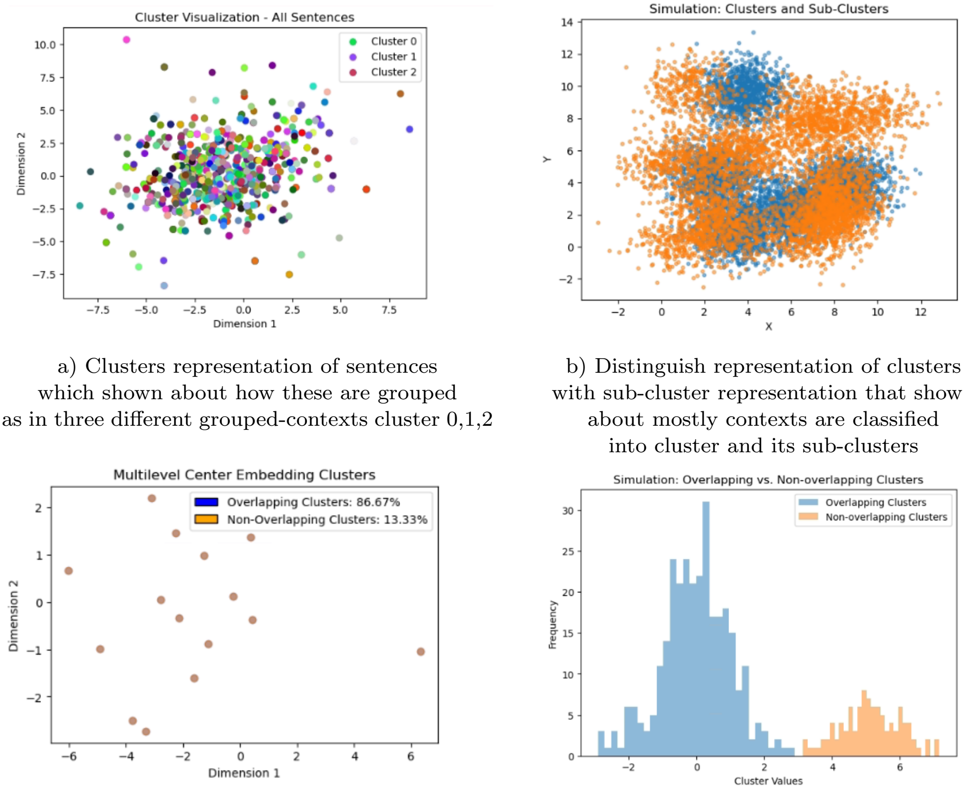
Figure 5. Examined 5 random Cases from Semcor corpus: Individual sense predictions of successfully captured senses which represented as [Contexts: captured sentence from corpus, Ambiguous word, Predicted Senses, Known Senses: ground truth values from sense-directory, with effective precision, recall, F1-score]

After getting such effective score over SemCor corpus, we have also performed some other demonstrations which are shown and discussed in following sub-sections to get in-depth analysis of proposed idea over WSD using multilevel clustering upto level-1 & 2 processing.

5.2. Demonstration of clustering results

As according to proposed idea where the clustering of word embedding based on the k-means clustering algorithm. Each point in the plot represents a word context, and the color of the point corresponds to the predicted sense of that word context. Figure 6 give some remarks over here as it show clear boundaries between clusters, indicating that the clustering process effectively groups word contexts with similar meanings together. Different clusters should contain points in Figure 6a and 6b, representing various word senses, showing that the WSD method captures distinct word senses and assigns them to separate clusters. Overall If clusters significantly overlap, it might suggest ambiguity in word senses or challenges in disambiguation, which could indicate why here is necessary to incorporate multilevel clustering.

With respect to the overlapping nature of the clusters as Figure 6c and 6d, it categorized the points into two groups: overlapping clusters and non-overlapping clusters based on the k-means clustering results. Overlapping clusters suggest that multiple word senses are assigned to the same cluster, indicating potential polysemy or ambiguous contexts. A higher overlapping percentage might indicate challenges in disambiguating word senses, while a lower percentage suggests successful separation of senses into distinct clusters.



a) Clusters representation of sentences which shown about how these are grouped as in three different grouped-contexts cluster 0,1,2

b) Distinguish representation of clusters with sub-cluster representation that shown about mostly contexts are classified into cluster and its sub-clusters

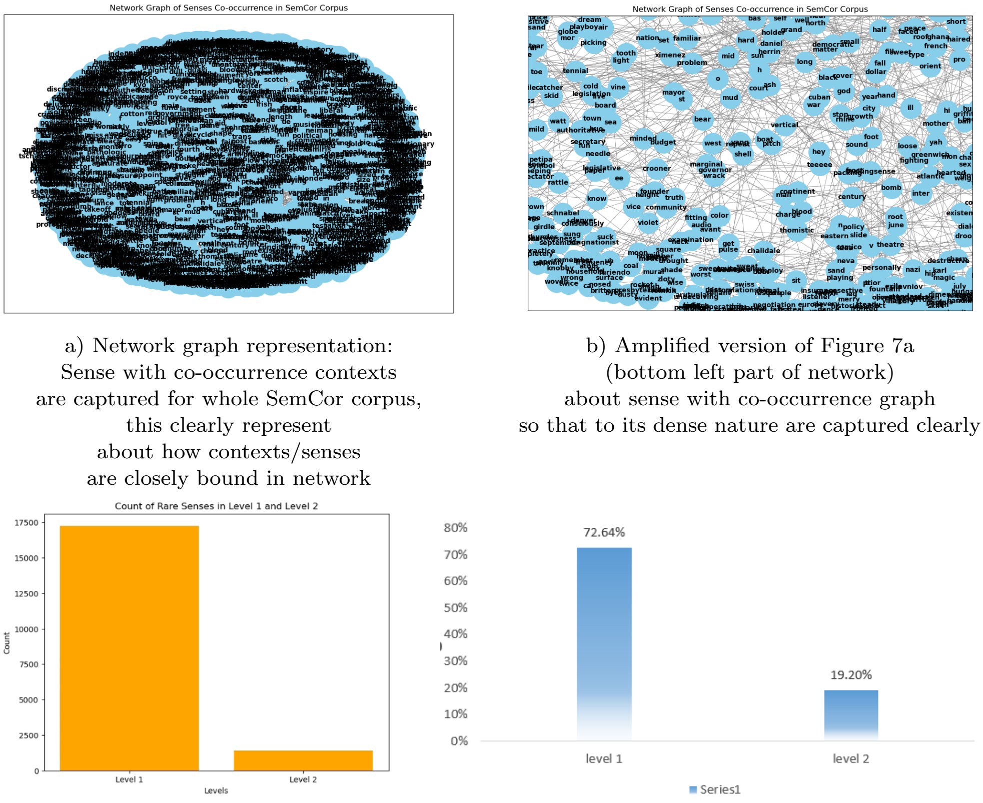
c) When center embedding applied over these clusters: 86% clusters are overlapping which help in performance of proposed WSD task, non-overlapping 13% score is depicting about limitation of proposed methodology

d) Another view of Figure 6c through bar chart which more clear representation of overlapping score among clusters

Figure 6. Clustering demonstration and Some key findings are observed using a, b, c, d sub-Figures: It also give support and insist the possibility of why chosen multilevel clustering upto the cluster with level 1 and 2

5.3. Distribution of senses in corpus as captured through proposed formulation

For the demonstration of senses distribution, in which shows relationship in co-occurrence, how much senses distributed in levels and what these senses connected to each others in the SemCor corpus. It helps to understand which senses are more prevalent and how they appear in the context. This information is crucial for understanding the overall sense usage in the corpus and identifying dominant senses or those that occur more frequently. The co-occurrence network graph as Figure 7a ang Figure 7b show how different senses are related to each other based on their co-occurrences in the SemCor corpus. It helps to identify senses that tend to co-occur frequently, indicating that they are semantically related or used together in similar contexts. As proposed initially to cover the aspect of polysemy, this visualization where a sense co-occurs with multiple other senses, it may indicate polysemy, where a word has multiple related senses. With the help of Figure 7d, where findings as with level-1 having approx 72% senses presence while in level-2 value is approx 19% as with proposed idea. In order to execute WSD utilizing multilevel clustering, it was shown that the majority of senses could be captured in level 1 in order to satisfy the greatest number of senses through connected contexts, while level 2 could considerably improve the performance score of the proposed technique for the situations of existing the majority of rare senses as in Figure 7c.



a) Network graph representation:
Sense with co-occurrence contexts
are captured for whole SemCor corpus,
this clearly represent
about how contexts/senses
are closely bound in network

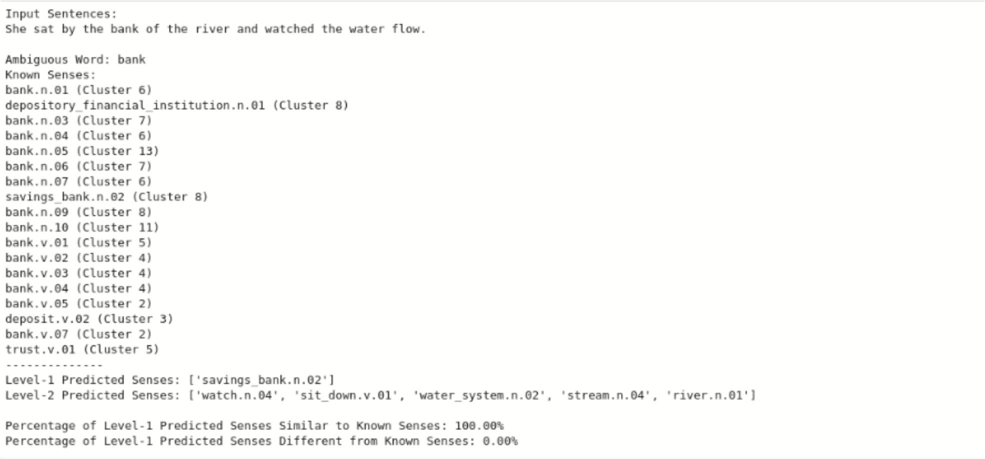
b) Amplified version of Figure 7a
(bottom left part of network)
about sense with co-occurrence graph
so that to its dense nature are captured clearly

c) Predicted rare-senses score
in leveled manner

d) Score of predicted senses in level-1 and level-2

Figure 7. To represent various relationship among senses with it's co-occurred senses to each other & capture senses in level 1 and 2

Additionally we have demonstrated our proposed method, by taking following example with input sentence for capturing predicted senses on the basis of similar context to input sentence and ambiguous word as Figure 8.



```
Input Sentences:
She sat by the bank of the river and watched the water flow.

Ambiguous Word: bank
Known Senses:
bank.n.01 (Cluster 6)
depository_financial_institution.n.01 (Cluster 8)
bank.n.03 (Cluster 7)
bank.n.04 (Cluster 6)
bank.n.05 (Cluster 13)
bank.n.06 (Cluster 7)
bank.n.07 (Cluster 6)
savings_bank.n.02 (Cluster 8)
bank.n.09 (Cluster 8)
bank.n.10 (Cluster 11)
bank.v.01 (Cluster 5)
bank.v.02 (Cluster 4)
bank.v.03 (Cluster 4)
bank.v.04 (Cluster 4)
bank.v.05 (Cluster 2)
deposit.v.02 (Cluster 3)
bank.v.07 (Cluster 2)
trust.v.01 (Cluster 5)
--------------
Level-1 Predicted Senses: ['savings_bank.n.02']
Level-2 Predicted Senses: ['watch.n.04', 'sit_down.v.01', 'water_system.n.02', 'stream.n.04', 'river.n.01']

Percentage of Level-1 Predicted Senses Similar to Known Senses: 100.00%
Percentage of Level-1 Predicted Senses Different from Known Senses: 0.00%
```

Figure 8. An application of proposed method which shown the working of how to predicted senses at leveled manner

In the sentence where in respect to take input with ambiguous word and find the range of known senses from WordNet hierarchy, now as proposed algorithm, find predicted senses in leveled manner where level-1 reflected the results as similar to known senses while on the other hand, when algorithm worked over level-2 it also predicted some other senses based upon the similar context to input sentence. The results are getting very attractive and meaningful at both level-1 and 2.

6. Conclusion and future directions

As proposed the hybrid methodology of word sense disambiguation (WSD) that utilizing multilevel clustering along with center-embedding based context embedding, it has demonstrated promising results in capturing various senses of ambiguous words in related contexts. The strategy grouped context terms into clusters using word/center embedding and multilevel (level-2) k-means clustering, which enables the system to speculate on possible meanings for the ambiguous word. Comparatively, the model is shown to be capable of accurately identifying pertinent senses in context, as seen by the performance evaluation's in Table 2. The system's ability to handle words with numerous meanings is shown by the polysemy capture rate of 29.6%, with low homonymy-capture rate of 2.3% implies that the model can distinguish between words with similar forms but different meanings as Figure 9.

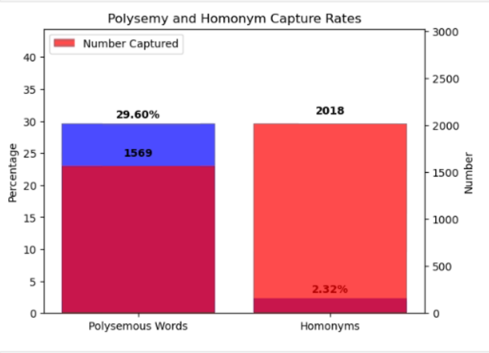


Figure 9. Polysemy and Homonym captured by proposed method

Table 2
Performance comparison of existing vs. proposed methods for Word Sense Disambiguation

Existing methods	Coverage [%]	Precision [%]	Recall [%]	F1-score [%]
[19]	100	45.4	45.4	–
[3]	100	–	–	70
[13]	–	–	–	49.8
[5]	–	66.5	64.8	65.7
[9]	100	–	–	79
As Proposed method	100	88.6	88.13	**88.36**

6.1. Limitations and future scope

Despite having achieved encouraging results, there are following rooms available of possibilities of improvements.

- In the view of Fine-Grained Sense Disambiguation where by grouping words at various levels, future research can concentrate on streamlining the clustering procedure to get a more precise word meaning disambiguation.
- In the case of contextual embedding, furhter go through with deep-learning strategies, where to utilize Bi-LSTM to handle and catpured more effective context-embedding (through treatment over complex sentences) which effectively worked over word senses hopefully.
- As the results of polysemy, the model demonstrated effectiveness in capturing a word's various meanings (polysemy), but in case of homonymy where score not observed effectively. Further research may look at methods to better handle words with similar forms but dissimilar meanings (homonymy).

When using the suggested multilevel clustering to further investigate other types of relations, such as homographs, it is hoped that the definition of a homograph would be developed to allow for the exploration of further levels while leveraging the definitions of WordNet-synsets.

Acknowledgements

With great appreciation, We would like to thank the reviewers whose insightful comments have greatly improved the quality of this article. Additionally, to express our gratitude to the Editor, along with the team-members by providing suggestions in helping to shape this work into its finished form.

Lastly, I also want to express my sincere gratitude to my son (Krishang), whose new outlook and astute insights gave me important clues for locating and resolving ambiguity issues in the texts. His advice has been very helpful in maintaining connection and clarity throughout to develop proposed algorithm.

References

[1] Agirre E., López de Lacalle O., Soroa A.: Random Walks for Knowledge-Based Word Sense Disambiguation, *Computational Linguistics*, vol. 40(1), pp. 57–84, 2014. doi: 10.1162/COLI_a_00164.

[2] AlMousa M., Benlamri R., Khoury R.: A novel word sense disambiguation approach using WordNet knowledge graph, *Computer Speech & Language*, vol. 74, 101337, 2022. doi: 10.1016/j.csl.2021.101337.

[3] Anaya-Sánchez H., Pons-Porrata A., Berlanga-Llavori R.: TKB-UO: Using Sense Clustering for WSD. In: *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 322–325, 2007. doi: 10.3115/1621474.1621544.

[4] Berahmand K., Li Y., Xu Y.: A Deep Semi-Supervised Community Detection Based on Point-Wise Mutual Information, *IEEE Transactions on Computational Social Systems*, vol. 11(3), pp. 3444–3456, 2023. doi: 10.1109/TCSS.2023.3327810.

[5] Bhingardive S., Singh D., Rudramurthy V., Redkar H., Bhattacharyya P.: Unsupervised most frequent sense detection using word embeddings. In: *Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*, pp. 1238–1243, 2015. doi: 10.3115/v1/n15-1132.

[6] Chifu A.G., Hristea F., Mothe J., Popescu M.: Word sense discrimination in information retrieval: A spectral clustering-based approach, *Information Processing & Management*, vol. 51(2), pp. 16–31, 2015. doi: 10.1016/j.ipm.2014.10.007.

[7] Dubey S., Kohli N.: A Multilevel Center Embedding approach for Sentence Similarity having Complex structures. In: *2023 World Conference on Communication & Computing (WCONF)*, pp. 1–8, 2023. doi: 10.1109/wconf58270.2023.10235102.

[8] Guerrieri A., Rahimian F., Girdzijauskas S., Montresor A.: Tovel: Distributed graph clustering for word sense disambiguation. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 623–630, IEEE, 2016. doi: 10.1109/icdmw.2016.0094.

[9] Huang L., Sun C., Qiu X., Huang X.: GlossBERT: BERT for word sense disambiguation with gloss knowledge. In: K. Inui, J. Jiang, V. Ng, X. Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3509–3514, 2019. doi: 10.18653/v1/d19-1355.

[10] Le A.C., Shimazu A., Huynh V.N., Nguyen L.M.: Semi-supervised learning integrated with classifier combination for word sense disambiguation, *Computer Speech & Language*, vol. 22(4), pp. 330–345, 2008. doi: 10.1016/j.csl.2007.11.001.

[11] Li X., Qing S., Zhang H., Wang T., Yang H.: Kernel methods for word sense disambiguation, *Artificial Intelligence Review*, vol. 46, pp. 41–58, 2016. doi: 10.1007/s10462-015-9455-5.

[12] Lyashevskaya O., Mitrofanova O., Grachkova M., Romanov S., Shimorina A., Shurygina A.: Automatic Word Sense Disambiguation and Construction Identification Based on Corpus Multilevel Annotation. In: *Text, Speech and Dialogue: 14th International Conference, TSD 2011, Pilsen, Czech Republic, September 1–5, 2011. Proceedings 14*, pp. 80–90, Springer, 2011. doi: 10.1007/978-3-642-23538-2_11.

[13] Martín T., Berlanga-Llavori R.: A clustering-based approach for unsupervised word sense disambiguation, *Procesamiento del Lenguaje Natural*, vol. 49, pp. 49–56, 2012.

[14] Miller G.A., Chodorow M., Landes S., Leacock C., Thomas R.G.: Using a semantic concordance for sense identification. In: *HLT '94: Proceedings of the workshop on Human Language Technology*, 1994. doi: 10.3115/1075812.1075866.

[15] Niu Z.Y., Ji D.H., Tan C.L.: Learning model order from labeled and unlabeled data for partially supervised classification, with application to word sense disambiguation, *Computer Speech & Language*, vol. 21(4), pp. 609–619, 2007. doi: 10.1016/j.csl.2007.02.001.

[16] Pasini T., Scozzafava F., Scarlini B.: CluBERT: A cluster-based approach for learning sense distributions in multiple languages. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4008–4018, 2020. doi: 10.18653/v1/2020.acl-main.369.

[17] Patil A.P., Ramteke R., Bhavsar R., Darbari H.: Graph-Based Algorithm for Word Sense Disambiguation: A Performance and Comparison, *Sambodhi*, vol. 44(03), pp. 77–79, 2021.

[18] Pelevina M., Arefyev N., Biemann C., Panchenko A.: Making sense of word embeddings, *arXiv preprint arXiv:170803390*, 2017. doi: 10.48550/arXiv.1708.03390.

[19] Seo H.C., Chung H., Rim H.C., Myaeng S.H., Kim S.H.: Unsupervised word sense disambiguation using WordNet relatives, *Computer Speech & Language*, vol. 18(3), pp. 253–273, 2004. doi: 10.1016/j.csl.2004.05.004.

[20] Shirai K., Nakamura M.: JAIST: Clustering and classification based approaches for Japanese WSD. In: K. Erk, C. Strapparava (eds.), *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 379–382, Association for Computational Linguistics, Uppsala, Sweden, 2010. https://aclanthology.org/S10-1085.

[21] Vaishnav Z.B., Sajja P.S.: Knowledge-Based Approach for Word Sense Disambiguation Using Genetic Algorithm for Gujarati. In: S.C. Satapathy, A. Joshi (eds.), *Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies*, vol. 1, pp. 485–494, Springer, Singapore, 2019. doi: 10.1007/978-981-13-1742-2_48.

Affiliations

ShivKishan Dubey
Harcourt Butler Technical University, Kanpur, skd@aith.ac.in

Narendra Kohli
Department of Computer Science, HBTU Kanpur, Uttar Pradesh, nkohli@hbtu.ac.in

Received: 19.10.2023
Revised: 06.02.2024
Accepted: 20.02.2024

Sanjai Pathak, Ashish Mani,
Mayank Sharma, Amlan Chatterjee

QUANTUM INSPIRED CHAOTIC SALP SWARM OPTIMIZATION FOR DYNAMIC OPTIMIZATION

Abstract *Many real-world problems are dynamic optimization problems that are unknown beforehand. In practice, unpredictable events such as the arrival of new jobs, due date changes, and reservation cancellations, changes in parameters or constraints make the search environment dynamic. Many algorithms are designed to deal with stationary optimization problems, but these algorithms do not face dynamic optimization problems or manage them correctly. Although some optimization algorithms are proposed to deal with the changes in dynamic environments differently, there are still areas of improvement in existing algorithms due to limitations or drawbacks, especially in terms of locating and following the previously identified optima. With this in mind, we studied a variant of SSA known as QSSO, which integrates the principles of quantum computing. An attempt is made to improve the overall performance of standard SSA to deal with the dynamic environment effectively by locating and tracking the global optima for DOPs. This work is an extension of the proposed new algorithm QSSO, known as the Quantum-inspired Chaotic Salp Swarm Optimization (QCSSO) Algorithm, which details the various approaches considered while solving DOPs. A chaotic operator is employed with quantum computing to respond to change and guarantee to increase individual searchability by improving population diversity and the speed at which the algorithm converges. We experimented by evaluating QCSSO on a well-known generalized dynamic benchmark problem (GDBG) provided for CEC 2009, followed by a comparative numerical study with well-regarded algorithms. As promised, the introduced QCSSO is discovered as the rival algorithm for DOPs.*

Keywords computational intelligence, swarm intelligence, salp swarm algorithm, dynamic optimization, quantum computing

Citation Computer Science 25(2) 2024: 301–326

Nomenclature

w_t – value of the chaotic map at the t-iteration
z_n – part of a generic superposition for n-qubits
X^i – position of i^{th} Salp
x_j^i – position of i^{th} Salp in j^{th} dimension
A_j – local attractor for the convergence speed in the search space j^{th} dimension
B_l – known as contraction-expansion coefficient in l^{th} generation
u_d – chaotic operator equation in d^{th} dimension
F_n – n^{th} benchmark function
T_n – n^{th} change type

1. Introduction, motivation

Meta-heuristic methods are popular for solving complex optimization problems, an intelligent approach in which an iterative process enhances the obtained solution until a concluding state is achieved. Most meta-heuristic methods are designed and implemented to work on static optimization problems, where the search space and the problem environment remain unchanged. The algorithm strived to achieve global optima during optimization [24]. But, most real-world optimization problems nowadays are dynamic and stochastic optimization problems, where the problem of domain space changes throughout the optimization process, and the obtained solution is not more relevant after environment changes. For example, the gantry crane scheduling task is a problem that is generally determined as a stationary optimization. But, entering another job throughout the planning process or the failure of a gantry crane, the search area environment changes from static to dynamic. The previously obtained solution may no longer apply to the new problem space. This kind of problem is known as a dynamic optimization problem in works of literature [3, 22].

The discovery of global optimization is the primary objective while solving stationary optimization problems, but finding only global optimal is insufficient for dynamic optimization problems. However, identifying and tracking global optimal is critical for DOPs in dynamic environments. The optimization methods designed for static optimization problems do not follow the optimal appropriately. Therefore, such techniques are not appropriate for dynamic optimization problems. It is essential to find the different methods that align with the objective of dynamic optimization problems and other assessment benchmarks for optimization in dynamic and uncertain environments.

The DOP solution (i.e., defined as S), which constantly evolves through the optimization method, influences the execution of many real-world applications [15].

$$S = z(y,\ \varnothing,\ t)$$

when S must be the dynamic optimization problem specified as cost function z inclusive y as a feasible solution from the set of solution Y, $\varnothing$ is the control parameter to find the solution distribution in the fitness landscape, and t is the time. The search strategy of the algorithms must be efficient to localize and follow the evolution of the global optima in time t toward finding superior solutions in the fitness landscape for DOPs. Several techniques were suggested in the bibliography to deal with dynamic optimization problems. Meta-heuristic methods were used frequently, including swarm intelligence. The schemes which are used with meta-heuristic methods for the dynamic optimization problems are diversity schemes [10, 11], memory schemes [3, 39], multi-population schemes [4, 27], adaptive schemes, [37, 38], multiobjective optimization for dynamic environments [5] and an adaptive quantum-inspired evolutionary algorithm (AQiEA) for optimizing power losses by dynamic load allocation on distributed generators [19].

Particle Swarm Optimization (PSO) is an approach of optimization based on swarm intelligence, a population-based stochastic optimization approach. PSO is inspired by the social behavior of flocking birds and the schooling of fish. Like other evolutionary algorithms, PSO has outperformed in solving many real-world static optimization problems [24]. However, DOPs are difficult for the standard PSO to solve due to outdated memory when the environment change and diversity loss. A clustering PSO (CPSO) is introduced to address the critical issues of PSO for DOPs, with a local search strategy and clustering method for locating and following multiple optima in the dynamic environment [14]. Many researchers have reported the efficiency of quantum-inspired evolutionary algorithms (QiEA) for solving several combinatorial and benchmark problems in a variety of areas, including Engineering optimization problems [18], Data clustering [7], and Image Processing [29]. In a similar thought, we attempt to extend the standard SSA for the DOPs known as QSSO [27], which is based on a set of methods, including quantum computing, multi-population, and an intelligent shifting operator, to effectively explore the search space during the optimization process in dynamic environments. Further, this paper details the approaches considered while solving DOPs. The generalized dynamic benchmark problem (GDBG) provided for CEC'09 has been employed to assess the presented QCSSO, a well-known standard benchmark problem for evaluating optimization algorithms in dynamic environments.

Salp Swarm Algorithm (SSA) was introduced by Mirjalili et al. in 2017. SSA is a population-based metaheuristic optimization technique that impersonates the swarming behavior of Salp in the ocean by establishing a Salp chain [20]. Several conducted research mainly related to the advancement of SSA for solving real-world problems. A simple SSA with a random search radius was submitted to improve the proficiency of SSA [34]. A particle-based approach for SSA with global exploring and local exploiting was introduced for convergence speed and accuracy [33]. Hybrid SSA with a gravitational search algorithm was studied in [16] to boost its searchability. An elite-based SSA is introduced for numerical optimization problems by improving the searchability of the algorithm [26].

Further, SSA is widely employed in various engineering fields, such as controller placement problems [25] and multilevel color image segmentation [35]. All these applications have demonstrated the pertinency and efficiency of the SSA. The SSA is similar to other evolutionary algorithms in many characteristics and works effectively for many real-world applications. The swarming behavior of Salp in SSA can avoid converging each solution into a local optimum up to a few degrees due to the Salp chain [2]. But there are many optimization problems in the real world where it is difficult for the standard SSA to work efficiently, and sometimes it fails to optimize. Dynamic optimization problems (DOPs) are another domain where optimum global changes over time and SSA fails to improve the obtained global best solution to accomplish the expected global optima in the dynamic search space. The problem lies in SSA for DOPs primarily because of not having a good search scheme and loss of population diversity, when this must boost the global best solution obtained so far to achieve the expected global optimum. The standard SSA search strategy was designed to achieve global optima for static optimization problems. It cannot work as expected for problems in dynamic and uncertain environments.

For DOPs, SSA needs to improve with a good search strategy where it is required to enhance the obtained global best solution and improve the population diversity to prevent stagnation in local optima. A multi-population mechanism in QCSSO is employed to locate and track the multiple local optima throughout the optimization procedure. Further, this paper examines and compares the performance of QCSSO with well-regarded algorithms QSSA, Standard Particle Swarm Optimization (SPSO), and Clustering Particle Swarm Optimization (CPSO). However, SSA has been evaluated in several applications, especially on engineering design optimization problems, but hardly ever employed on DOPs, as seen from the survey [1].

The paper is structured as follows: Section 2 presents the relevant related work. Section 3 presents the proposed algorithm QCSSO with the techniques used for DOP, and Section 4 presents the experiment configurations. Experimental evaluation and discussion are presented in Section 5. Finally, the conclusion and relevant future work are presented in Section 6.

2. Related work

There are specific challenges for the meta-heuristic methods in dynamic environments during the optimization process that are not there for static optimization problems: (1) A good search strategy for DOPs to locate and track global optimum, (2) Outdated memory, (3) Diversity loss. In the dynamic environment, the fitness of the attained solutions changes due to the dynamic environments. It will no longer coincide with the retained value in the memory exploited by the algorithm. Loss of diversity occurs in every meta-heuristic method due to an intrinsic nature of convergence to the global optima. Generally, the loss of diversity reason is the built-in property of meta-heuristic algorithms for convergence to the previous optimal position and the excessive vicinity of the solutions to each other.

The re-initialization is a straightforward way to prevent convergence on the previous optimum stance and excessive closeness of the solutions [13]. In re-initialization, the method considers the changing environment as a new optimization problem and relaunches the optimization process using the evolved environment. However the efficiency of the optimization algorithm could improve upon the acquired knowledge of the earlier environment, and the re-initialization technique insinuated the depletion of all obtained knowledge so far from the search space. Although outdated memory is of less relevance in comparison to diversity loss, there are solutions proposed in the literature to handle it [6]: (1) Forgetting memory, (2) Re-evaluating memory. These two proposed solutions apply to the optimization approaches where the obtained information from the search space is stored. In the forgetting memory technique, the position preserved for each solution will be substituted by the positions of the new environment. In the re-evaluating memory technique, the saved positions in memory are re-evaluated.

The diversity loss is because of the built-in character of the meta-heuristic technique, designed originally for static optimization, where quick convergence has been considered a good feature. The works of literature proposed several solutions, including a memory-based approach, to store previous optimal solutions to use when the environment changes, mutation, and self-adaption in which diversity loss is allowed and later to solve it for the expected outcome of the optimization algorithm. An adaptive mutation operator, i.e., activated Hyper-Mutation, was suggested as a factor to be multiplied by the specific mutation to create diversity [9]. In [21], an adaptive approach is employed for the chaotic mutation to create diversity in the environment. Another technique was introduced as a local variable search to address the consistency problem in the mutation step size by making it adaptive [32]. In [39], replacing random solutions with formerly obtained solutions in case of environmental changes is an alternative suggestion proposed to produce diversity. In [36], an implicit memory technique presents diploid genetic algorithms for dynamic optimization.

Multi-population is another approach in uncertain and dynamic environments, considered a hybrid model to create and maintain diversity for the DOPs. In this technique, the population is divided into subpopulations, covering various search space regions. In general, all the subpopulations have similar tasks but can also have distinct tasks. In [23], an approach based on a genetic algorithm was proposed as the Shifting Balance Genetic Algorithm (SBGA). In SBGA, many subpopulations are in charge of global search, and a large subpopulation monitors the evolving peaks. In [4], another approach based on self-organizing scouts is proposed, termed Self-Organizing Scouts (SOS). The SOS conversely utilizes a large subgroup for global search and a few subpopulations for monitoring changes. This policy has also been suggested with other optimization algorithms, such as Genetic Algorithm [8]. In [17] a procedure for optimization in dynamic environments is presented as compound particles. This procedure has an agreeable throughput, and in [40] a new method-built artificial fish swam algorithm to show a similar approach.

Like any other evolutionary algorithm, quick convergence is a good feature in SSA and benefits many static optimization problems. The fast convergence is mainly due to the loss of diversity in the population, and control parameters are fixed during the optimization process. In the SSA, all the Salp attracted strongly towards food position and sometimes converged on local or global optima where food position is located. However, because of this feature, the original SSA considers the optimization problem a single instance problem and does not adapt when the environment changes. The result is unsuitable for DOPs, where a change reflects an entirely new problem, in which locating and tracking the evolving global optima in the search landscape is necessary. Standard SSA needs to ameliorate by adapting the different mechanisms to control diversity. Controlling diversity is an excellent strategy for DOPs that support converting good local optima into global best in the changing environment during optimization.

3. The Quantum Inspired Chaotic Salp Swam Optimization Algorithm

A good exploring and exploitation tendency of the SSA heuristic on static optimization problems enables it to attract the DOPs. The standard SSA has the superiority of specific updating functions but still has the problem of quick convergence due to loss of diversity and readily falls into the local optimum, particularly when encountering multidimensional and dynamic optimization. Motivated by the thought of implementing the quantum computing technique for metaheuristics to stimulate their global optimization achievement, this paper proposes Quantum Inspired Chaotic Salp Swarm Optimization (QCSSO) Algorithm with a multi-population mechanism to locate and track the optima. Implementing a chaotic series based on a chaotic logistic map in place of a random series is a sound approach to diversifying the population and enhancing the QCSSO execution in prevention from early convergence to local optima. The fundamentals of quantum computing and the detailed QCSSO process are developed in this section.

3.1. Quantum computing

Quantum computing (QC) is an emergent and novel approach to computational intelligence. It is advanced in the concepts and principles of quantum physics. This integration emerged from the knowledge of quantum computers, where specific calculations are carried out more quickly than digital computers. An accelerated analysis is made possible using the quantum principles of computation, such as the superposition of states, entanglement, and interference [7, 29]. In a quantum computer, a particle can be in a superposition state where two or more quantum states can be added together, resulting in another valid quantum state. Unlike bit (0 or 1) in a digital computer, a quantum bit or qubit is a quantum computer's tiniest data unit. The qubit can be in primary states represented as Dirac notation using $|0\rangle$, $|1\rangle$ or linear superposition states of $|0\rangle$ and $|1\rangle$.

A physical system can be in one of many arrangements of particles or fields. According to the quantum superposition principle, the state combines all these possibilities. The qubit representation is probabilistic and is defined as a pair of numbers $(\alpha,\ \ \beta)$ for two possible arrangements, 0 and 1, of particles. Equation (1) describes the physical system as a qubit state [27].

$$|\psi\rangle = \alpha\ \ |0\rangle +\ \ \beta\ \ |1\rangle \ldots \tag{1}$$

where $|0\rangle$ and $|1\rangle$ are two basic states, the coefficients $\alpha,\ \ \beta$ are complex numbers with $|\ \alpha^2|\ +\ |\ \beta^2| = 1$ and dictates the probabilities of the system to be in either arrangement. Module $|\ \alpha\ |^2$ indicates the probability of the qubit being at state 0, and $|\ \beta\ |^2$ is the probability of the qubit being at state 1. The two basic states $|0\rangle$ and $|1\rangle$ are called computing base states of quantum bits, and they correlate to the two states 0 and 1 of digital computer bits. The notable difference between digital computer bits and qubits is that the qubits can be in a superposition state of $|0\rangle$ and $|1\rangle$ as presented in Equation (1).

Quantum entanglement presents a physical phenomenon of a system where a quantum state must not be factored as a product of states since the individual component is incomplete and cannot be described without considering the other features. Moreover, they are not individual particles but are inseparable wholes. The composite system state can always be expressed as a sum or superposition [12]. In quantum computer science and information processing, entanglement is a valuable physical resource and a prominent feature of multiple qubit systems. It is simple to realize that plenty $(x+y)$ qubit states couldn't be formulated as the tensor product of an x qubit state and an y qubit state as these are entangled states. For example, the entangled states that are maximally entangled (Bell states) and weakly entangled states are the same as $|00\rangle +\ \ 0.01\ \ |11\rangle$, also the separable ones such as $|00\rangle$.

Another concept and principle of quantum physics is interference, in which particles can be in more than one place at any given time using superposition and cross their trajectory to interfere with the direction. To realize the quantum interference, examine a generic superposition for n-qubits $\sum_n z_n\,|n\rangle$. The direct measure of z_n, only returns the local information for the possible values of n. However, the return results get changed once a unitary transformation is performed using:

$$|n\rangle = \sum_y p_{ny}|y\rangle \text{ for all } n$$

on it, and then,

$$\left(\sum_n z_n\,|n\rangle\right) = \sum_n z_n \left(\sum_y p_{ny}\,|y\rangle\right)$$
$$= \sum_y \left(\sum_n z_n p_{ny}\right)|y\rangle$$

If now we are measuring $(\sum_n z_n\,|n\rangle)$, global information for all z_n can obtain through amplitude $\sum_n z_n p_{ny}$ for a single value of y.

3.2. Logistic chaotic map

The chaotic logistic map is optimal from the tenfold adapted chaotic maps in [28]. We embedded chaos theory in the search strategy of SSA. We obtained the superior average value of the optimal solution with better balance corresponding to the original SSA and another meta-heuristic algorithm.

Chaos theory is a prominent mathematical strategy and has been used extensively in the literature to improve the performance of meta-heuristic algorithms. It is typically outlined as the simulator of the dynamic behavior of a non-linear scheme. A chaotic logistic map is employed with a genetic algorithm for encrypting the image. It is utilized to encrypt the early version of the image, and later GA is executed to boost the encryption results. A hybrid global optimization algorithm is proposed based on a chaos search strategy with a complex method to jump out from the local optima obtained.

We used the following equation in this article for the chaotic logistic map. The output of this equation is embedded into the equation of QCSSO, which prevents the problem of search stagnation:

$$w_{t+1} = d \cdot w_t \ (\ 1 - \ w_t) \ldots \tag{2}$$

where w_t is the value of the chaotic map at the t-iteration. In this article, the initial condition of the chaotic map is set to $(w_0 = 0.70)$.

3.3. The proposed algorithm

SSA simulates the swarming performance of Salp throughout the optimization and modeling of a Salp chain as shown in Figure 1.

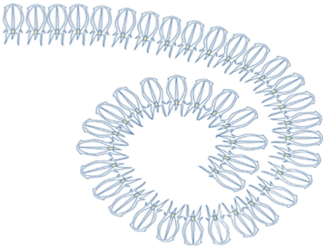


Figure 1. The Salp Chain

This chain may prevent stagnancy in the local optima up to a bit level because Salp is usually attracted towards the global optima by collaborating with leaders Salp. Also, the original SSA search process is ineffective in strengthening the obtained global best, i.e., to achieve global optima in the fitness landscape. The initial search process considers the optimization problem as a single problem instance. For DOPs, it is necessary to consider dynamic changes, respectively, such as the emergence of new cases of problems that must be tackled from scratch. Consequently, the standard SSA does not have the competence to examine and preserve the diversity of the population. Hence, it cannot solve dynamic optimization problems, where it is necessary to discover and monitor the global optima in dynamic and uncertain environments.

In this paper, we extended the work on a quantum-inspired algorithm based on SSA with a multi-population mechanism to discover and follow the global optima in the search space, an effective method for dynamic optimization problems (DOPs) [27]. The basic procedure and strategy of QCSSO are explained as follows:

- Multiple peaks exist in almost all dynamic environments, and each can transform into a global optimum when the environment changes. That means each peak can be a probable optimum. Hence, an optimization algorithm intended for DOPs should quickly track all the environmental peaks to identify the optimum peak on environmental change. The multi-population mechanism is a promising strategy that enhances the coverage of potential multiple optima in the search space [5].

- In the proposed multi-population strategy, a part of the population is detached from the total population to create a new sub-population. No information is being shared between the populations, excluding the duplication search among the two best individuals of two subpopulations to avoid multiple local best at a certain distance.

An approach to extend and change the original SSA with Quantum computing, multi-population mechanism, and the introduced chaotic operator could be a promising solution for the dynamic optimization problems. It accelerates the speed of SSA and supports locating and tracking the global optima, increasing the diversity of individuals and preventing stagnation in local optima in the search space.

In quantum science, the Delta potential well model describes using the Dirac delta function, which is a general function and objectively correlates to the potential that is zero universally except for a single point when it takes an infinite value. In the QCSSO, the Salp in the Delta potential well should move in the bound state in line with the strategy proposed for the particles in [30]. The position of Salp is necessary to be measured to assess the fitness of each Salp. However, only the probability of position for each Salp (X^i) can be find-out from the probability density function $|\Psi(x,\ t)|^2$, i.e., Salp emerges at position x relative to point A. Thus, it is necessary to measure Salp's position thanks to the collapsing methodology, i.e., transforming from a quantum state to a classical form. This measurement process can be simulated using Monte Carlo Method using the procedure mentioned in [30].

In this article, we use the following iterative equation for the position to measure each Salp according to the proposed equation in the quantum-inspired PSO for particles in [31]:

$$X_{j+1}^{k} = \begin{cases} A_d \ + \ B_l \cdot \left|BestMean_l \ - \ X_j^k\right| \cdot ln\left(\frac{r_d}{u_d}\right) & c_3 > 0 \\ A_d - \ B_l \cdot \left|BestMean_l \ - \ X_j^k\right| \cdot ln\left(\frac{r_d}{u_d}\right) & c_3 < 0 \end{cases} \quad \cdots \tag{3}$$

$$X_j^i = \left(X_j^{i-1} + C_i^{l\,+\,1}\left(A_j - X_j^i\right) + m \cdot (X_j^{i-1} - X_j^{i-2})\right)\ldots \tag{4}$$

$$u_d = 3 \cdot w \cdot (1 - w) \cdot c_4$$

where r_d, c_3, c_4 are uniformly generated random values inside [0, 1], and u_d is chaotic operator equation, x_j^i indicates the position of i^{th} Salp in j^{th} dimension, A_j reveals the local attractor for the convergence speed in the search space j^{th} dimension, B_l is known as the contraction-expansion coefficient, which gradually decreases through iterations. The coefficient $C_i^{l\,+\,1}$ is the most important parameter in the QCSSO, as presented in Equation (5) for the follower, Salp to support better coordination during the exploitation of the search space.

$$C_i^{l\,+\,1} = 0.75 \cdot \sin\left(\frac{\pi}{4}\right) \cdot \left(1 - \left(\frac{l}{L}\right)\right)\ \ldots \tag{5}$$

as l specifies the present iteration, L is the maximum number of iterations. We assumed here the first iteration $l = 0$ with maximum iteration size L. B_l describes as Equation (6), a contraction-expansion coefficient that gradually decreases or increases iteration-wise in line with the progress of respective Salp's convergence speed and execution of the algorithm. A_d as in Equation (7), a base point for Salp to move around in the vicinity. In addition, to know as an inclining learning point for Salp to oscillate around. $BestMean_l$ describes mathematically as Equation (8), the mean of the individual best position. X_j^k is the k-th Salp in j-dimension and X_{j+1}^k is the new position of Salp.

$$B_l = \left(\frac{0.5 \cdot (L - l)}{l + 0.5}\right)\ldots \tag{6}$$

where l is the current iteration and L is the maximum number of iterations.

$$A_d = \frac{\left(r_{1d} \cdot X_j^k + r_{2d} \cdot F_j\right)}{r_{1d} + r_{1d}}\ldots \tag{7}$$

where r_{1d} and r_{2d} are uniformly distributed random numbers in the range [0, 1]. F_j is the food position as the best location.

$$BestMean_l = \frac{1}{N}\sum_{j\,=\,1}^{d} x_{k,\ j}(l)\ldots \tag{8}$$

where N is the maximum number of the population.

3.4. Methodology

A few concepts are suggested in papers to enhance the efficiency of SSA. Still, we employed the standard SSA and implemented quantum theory, multi-population with a chaotic logistic operator as explained in the previous section, to boost the obtained optimal global solution by SSA and monitor the trajectory of the global optima.

Adapting the chaotic logistic operator supports preserving individuals' diversity and precluding re-initializing the population when the change is determined, as it offers an acute information depletion. Maintenance of diversity is essential for dynamic optimization as the global optimization of DOPs evolves through time. If the Salp is grouped in a narrow area, then the individual Salp cannot discover the changes in the problem. Also, the explorative ability of EAs is dependent and determined on the population diversity, which means the exploring ability is reduced in the identical elements of the population.

In QCSSO, Equation (3) and (4) updates the Salp positions and produces new solutions. The population is divided into subpopulations. An overlapping search technique between the two local best is employed to avoid entrapment in local optima and increase the searchability of the algorithm. The value of parameters $C_i^{l\ +\ 1}$ makes adjustments through the loop of the algorithm and produces new adapting values for the position to update throughout the optimization process. The quantities r_k k $\{1, 2, 3\}$ stand for evenly distributed random values within the range [0, 1]. The value of w was taken as a fixed value of 0.96.

Pseudo-code for the QCSSO presented in Algorithm 1 first initializes the population, the best position (i.e., food position), and other algorithm parameters.

Algorithm 1: Pseudo Code of QCSSO

```
method QCSSO(M, N, l, u, dim, rObj)
  INIT: P, R, fbest, ifitness, subp
  WHILE $N \leq M$ DO                *Main Loop
      Multi-population Strategy
      if $N \geq 1$ THEN
          Positions Update equation (3), (4)
          Values of Parameters update
          Fitness Calculation
      else
          Salp chain create using standard SSA [1]
      Evaluate local best in the subP
  end WHILE
Best Solution                         *Returns the Best Solution
```

Next, the Salp chain is built using the standard equation of SSA [24]. The Salp moves towards the optimal solutions using the quantum-inspired Equations of QCSSO (3) and (4), followed by the chaotic logistic mechanism to locate and track the global optima. A chaotic logistic map approach was applied while improving the position of Salp to maintain diversity. Since the nature of the SSA algorithm is iterative, it repetitively produces and develops some random Salp within the maximum and minimum limits of search space. Then, all the Salp, i.e., leader and followers, update their position in the location vector during optimization. A flow diagram of the QCSSO optimization algorithm, as displayed in Figure 2.

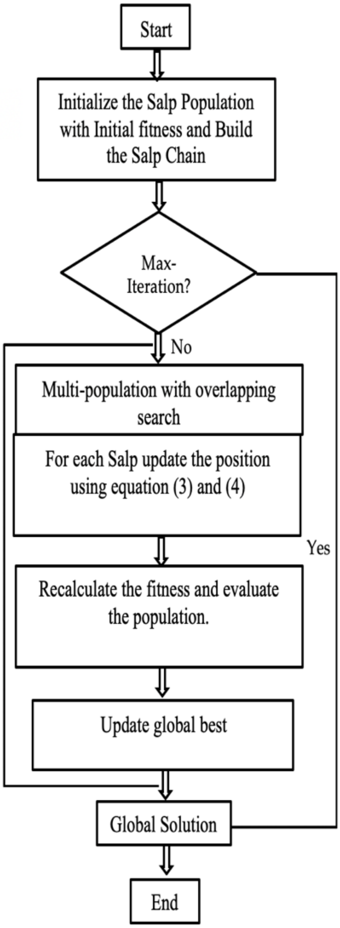

Start
Initialize the Salp Population with Initial fitness and Build the Salp Chain
Max-Iteration?
No
Yes
Multi-population with overlapping search
For each Salp update the position using equation (3) and (4)
Recalculate the fitness and evaluate the population.
Update global best
Global Solution
End

Figure 2. Flowchart of QCSSO optimization algorithm

3.5. Considerations for DOPs in SSA

To address the key issues of SSA for DOPs, such as quick convergence, and to maintain diversity in solution and population, the QCSSO approach with multi-population chaotic theory employ to locate and track the multiple local optima in search space. Multi-population is an effective and popular strategy used in the literature to facilitate the performance of evolutionary algorithms for dynamic optimization problems. Multi-population, where the entire population is divided into several subpopulations of small sizes to discover and monitor the constantly evolving global optima in a dynamic and uncertain environment. This strategy also helps control the solution's diversity and convert local optima into global best, which contributes to making the SSA search strategy more effective for DOPs. The initial assumption of local optima is considered in sub-population, and an overlapping search between the two best solutions is performed to avoid multiple convergences around the same point. No other information is shared between the populations. The superposition searching is performed among two subpopulations by comparing the best Salp in subpopulations. The attempt is to generate a new point around the best Salp in the subpopulations by adding a random value sampled from the normal distribution in the dimension loop.

Another mechanism to check the age of each Salp based on the number of generations has been considered for the Salp stagnates into local optima, which is an addition to the chaotic logistic map. The idea is to look for the Salp stagnates in local optima and re-initialize it for the next generation. According to the aging strategy, the global best is not considered. Only the local best whose age is more significant than a defined limit and no improvements during the specified number of iterations is considered to re-initialize as presented in Algorithm 2. The age of individual Salp increments by 1 in each iteration, and its fitness is evaluated to see improvement. In case of no improvements for several iterations, a flag is triggered to re-initialize it for the next generation.

Algorithm 2: Ageing approach

```
In the population loop
if i-th Salp is global best then do nothing
else if  i-th Salp is best in sub-population and bAge
         is greater than maxLimit and rRand less
         than 0.1
      re-initialize i-th Salp
else if  i-th Salp is not best but bAge is greater than
         minLimit and rRand less than 0.1
      re-initialize i-th Salp
until max-pop size
```

4. Experiments

Firstly, we introduce the experimental environment and basic test suites used in our experiments. Then, we define the parameter settings and evaluation mechanism for the algorithms. Finally, we discuss the state-of-the-art algorithms used for comparing results in the computer running environment.

4.1. Experimental context

A personal computer environment was prepared for the empirical study. The configuration is based on the following:

- Intel® Core™ i7-3520M CPU @ 2.90 GHz,
- Memory 16 GB,
- Operation System Microsoft Windows 10 Home,
- QCSSO's code is developed in CPP,
- GDBG Dynamic Framework in CPP.

4.2. Basic test functions

The achievement of the QCSSO optimization algorithm is assessed on the six standard functions (F) contemplated by Li *et al.* for the CEC'09 Special Session and Competition on "Evolutionary Computation in Dynamic and Uncertain Environments":

1. Peak Rotation Function (F1),
2. Sphere's Composition Function (F2),
3. Rastrigin's Composition Function (F3),
4. Griewank's Composition Function (F4),
5. Ackley's Composition Function (F5),
6. Hybrid Composition Function (F6).

It's a generalized dynamic benchmark generator (GDBG), a dynamic simulation framework for this study, and includes six benchmark functions [4]. The characteristics of GDBG include seven change types for the control parameters – large and small step change, recurrent and recurrent with noise and dimensional shift, and random and chaotic change:

1. Small Displacement (T1),
2. Large Displacement (T2),
3. Gaussian Displacement (T3),
4. Logistic Function (T4),
5. A Periodic Displacement (T5),
6. A Periodic Displacement with Noise (T6),
7. Random with Changed Dimension (T7).

All the benchmark functions with their range are presented in Table 1.

Table 1
Details of the benchmark functions

Function Name	Function	DIM	Range	$f_{\min}$
Sphere's Function	$F(x) = \sum_{i=1}^{n} x_i^2$	10	$[-100, 100]$	0
Rastrigrin Function	$F(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	10	$[-5, 5]$	0
Weierstrass Function	$F(x) \quad = \sum_{i=1}^{n} \left(\sum_{k=0}^{k_{\max}} \left[a^k \cos\left(2\pi b^k \left(x_i + 0.5\right)\right)\right]\right) -$ $- n \sum_{k=0}^{k_{\max}} \left[a^k \cos\left(\pi b^k\right)\right]$	10	$[-0.5, 0.5]$	0
Griewank Function	$F(x) = \frac{1}{4000} \sum_{i=1}^{n} \left(x_i\right)^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	10	$[-100, 100]$	0
Ackley's Function	$F(x) \quad = -20 \exp\left(-0.2 \frac{1}{n} \sum_{i=1}^{n} x_i^2\right) -$ $- \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos\left(2\pi x_i\right)\right) + 20 + e$	10	$[-32, 32]$	0

4.3. Method to set parameters and testing

The QCSSO is designed to work with five parameters on the GDBG framework: M-Maximum Iteration, N-Current Iteration, Global Optima, l-Lower bound, u-upper bound, and rObj-Objective function reference. Rest all necessary parameters for the algorithms administered by another function of GDBG: the number of Salp and number of dimension. In this section, the achievement of QCSSO is contrast with SPSO [24], CPSO [14], and QSSA [29].

The dynamic variant of PSO algorithms is assessed on the dynamic optimization problems and submitted in CEC'2009 for the maximum population size of 50. We have also set the population size as 50 for the QCSSO while using the GDBG framework. For the problem function F_1, two test numbers of peaks, i.e., 10 and 50, are being used, and for the other problem functions F_2 to F_6, one test number of peaks, i.e., 10, was used.

The dynamic benchmark problem examination process is employed for all the benchmark problems of GDBG. An interface has been built to make all the necessary adjustments from the QCSSO execution context. The place of parameters is aligned in code to retain projected exploring and exploiting in the search space of DOPs. The QCSSO algorithm has been set to execute for the pre-defined number of assessments for the evaluation functions (F) and necessary parameters with change types (T) as input. It means no information is being shared during the execution of the QCSSO algorithm related to the problem change, including the number of peaks, dynamic, or dimension change.

4.4. Performance assessment of algorithm

The total 49 test cases of the six essential test functions (i.e., F1–F6) with seven change types (i.e., T1–T7) is considered for the performance assessment of the optimizer. We recorded the values of errors in best case average, the mean average, worst average, and STD for each possible case, which is defined as in [24]:

$$\text{Average-best} = \sum_{i=1}^{runs} Min_{j=1}^{num-change} \frac{E_{i,j}^{last}(t)}{runs}$$

$$\text{Average-mean} = \sum_{i=1}^{runs} \sum_{j=1}^{num-change} \frac{E_{i,j}^{last}(t)}{runs \cdot num - change}$$

$$\text{Average-worst} = \sum_{i=1}^{runs} Max_{j=1}^{num-change} \frac{E_{i,j}^{last}(t)}{runs}$$

here, $E_{i,j}^{last}(t) = |\ Q\ (y^{b}(\ \mathrm{t}\ )) - \mathrm{Q}\ (\mathrm{y} \cdot (\mathrm{t}))\ |$, i.e., to calculate for reaching max change for each change type and $y \cdot (t)$ is the global optimum at time t.

5. Discussion on experimental results

We conduct practical experiments with numerical evaluation to prove the theoretical claims undertaken in previous sections. Numerical results are gathered in the sum of marks obtained in each case and multiplied by 100 to measure the score in percentage. The original SPSO, QSSA, and QCSSO score is evaluated in the said environment, and reference score values of CPSO are as per the score obtained in [14]. This section presents the results of the experiments, including the comparison report with peer algorithms and analysis of different change effects for each function during the optimization process.

5.1. Experimental analysis

The QCSSO is implemented and assessed for dynamic optimization problems (DOPs). The measured value of execution on each problem is recorded in the form of best average, mean average, worst average, and standard deviation (SD), tabled in Table 2 to Table 8 as a result of the analysis. In Table 9, the score of each algorithm is lodged with all six benchmark problems and seven change types and with the combination of different test cases. Assessment results and data analysis show that QCSSO has performed better on most benchmark problems than SPSO, QSSA, and CPSO optimization algorithms.

Table 2

F1: result achieved on peak 10

	Errors	T1	T2	T3	T4	T5	T6	T7
QCSSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	2.01E − 02	3.23E + 01	3.24E + 01	6.04E − 09	2.52E + 01	6.16E + 01	2.90E + 01
	Avg. Mean	3.36E − 04	2.52E + 00	5.02E + 00	3.19E − 10	2.09E + 00	2.85E + 00	4.42E + 00
	STD.	3.68E − 03	6.86E + 00	9.65E + 00	1.50E − 09	4.65E + 00	1.04E + 01	7.82E + 00
QSSA	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	2.55E + 01	5.03E + 01	4.74E + 01	7.03E + 01	4.00E + 01	7.61E + 01	3.43E + 01
	Avg. Mean	4.04E + 00	8.56E + 00	1.68E + 01	1.70E + 01	7.09E + 00	2.87E + 01	3.80E + 00
	STD.	6.25E + 00	1.20E + 01	1.74E + 01	1.98E + 01	8.64E + 00	2.48E + 01	7.11E + 00
SPSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	2.50E + 01	4.89E + 01	4.81E + 01	7.55E + 01	2.20E + 01	8.04E + 01	4.17E + 01
	Avg. Mean	4.88E + 00	1.07E + 01	1.66E + 01	2.57E + 01	4.60E + 00	2.88E + 01	1.43E + 01
	STD.	6.74E + 00	1.31E + 01	1.78E + 01	2.06E + 01	5.78E + 00	2.59E + 01	1.37E + 01

Table 3

F1: result achieved on peak 50

	Errors	T1	T2	T3	T4	T5	T6	T7
QCSSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	5.81E + 00	2.97E + 01	4.01E + 01	8.32E − 01	8.38E + 00	4.51E + 01	2.80E + 01
	Avg. Mean	6.07E − 01	4.65E + 00	1.39E + 01	5.56E − 02	1.17E + 00	2.28E + 00	2.54E + 00
	STD.	1.50E + 00	7.16E + 00	1.11E + 01	2.06E − 01	1.83E + 00	7.34E + 00	5.08E + 00

Table 3 cont.

QSSA	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	1.42E − 14	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	2.43E + 01	4.20E + 01	4.11E + 01	6.13E + 01	2.08E + 01	7.54E + 01	3.14E + 01
	Avg. Mean	4.81E + 00	1.16E + 01	1.08E + 01	1.71E + 01	4.62E + 00	2.93E + 01	8.80E + 00
	STD.	5.98E + 00	1.08E + 01	1.01E + 01	1.59E + 01	5.42E + 00	2.59E + 01	7.85E + 00
SPSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	2.97E + 01	4.40E + 01	4.44E + 01	7.62E + 01	3.28E + 01	8.03E + 01	4.89E + 01
	Avg. Mean	7.68E + 00	1.21E + 01	1.70E + 01	2.10E + 01	5.68E + 00	3.97E + 01	1.37E + 01
	STD.	7.22E + 00	1.06E + 01	1.28E + 01	2.02E + 01	6.57E + 00	2.76E + 01	1.23E + 01

Table 4

F2: result achieved on peak 10

	Errors	T1	T2	T3	T4	T5	T6	T7
QCSSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	1.00E + 01	5.31E + 02	4.76E + 02	5.78E + 00	4.29E + 02	2.46E + 01	3.19E + 01
	Avg. Mean	7.59E − 01	1.76E + 01	2.79E + 01	3.61E − 01	3.80E + 01	2.71E + 00	5.12E + 00
	STD.	2.21E + 00	6.88E + 01	9.26E + 01	1.15E + 00	9.05E + 01	6.83E + 00	8.96E + 00
QSSA	Avg. Best	7.71E − 13	6.00E − 13	7.74E − 13	3.73E − 13	0.00E + 00	3.30E − 13	5.63E − 13
	Avg. Worst	7.42E + 01	5.69E + 02	5.28E + 02	6.22E + 02	4.94E + 02	9.78E + 01	4.72E + 02
	Avg. Mean	2.92E + 01	1.68E + 02	1.00E + 02	9.15E + 01	1.35E + 02	3.76E + 01	3.82E + 01
	STD.	1.99E + 01	2.22E + 02	1.71E + 02	1.41E + 02	1.73E + 02	3.22E + 01	6.97E + 01
SPSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	1.56E + 02	5.97E + 02	5.04E + 02	3.55E + 02	5.02E + 02	4.97E + 02	4.02E + 02
	Avg. Mean	4.59E + 01	1.71E + 02	1.63E + 02	5.36E + 01	1.88E + 02	7.56E + 01	4.84E + 01
	STD.	3.32E + 01	2.15E + 02	1.94E + 02	6.12E + 01	2.02E + 02	8.31E + 01	7.82E + 01

Table 5

F3: result achieved on peak 10

	Errors	T1	T2	T3	T4	T5	T6	T7
QCSSO	Avg. Best	0.00E + 00	4.32E − 11	0.00E + 00	1.45E − 13	5.25E − 01	0.00E + 00	0.00E + 00
	Avg. Worst	3.80E + 02	9.30E + 02	9.20E + 02	1.09E + 03	8.69E + 02	1.11E + 03	8.61E + 02
	Avg. Mean	3.53E + 01	6.19E + 02	4.96E + 02	1.19E + 02	4.84E + 02	2.89E + 02	2.03E + 02
	STD.	9.95E + 01	3.65E + 02	3.84E + 02	3.03E + 02	3.79E + 02	3.98E + 02	3.22E + 02
QSSA	Avg. Best	0.00E + 00	1.89E + 01	5.89E + 00	6.26E − 13	9.90E + 00	7.78E − 13	1.32E − 12
	Avg. Worst	8.93E + 02	1.06E + 03	9.84E + 02	1.34E + 03	9.84E + 02	1.42E + 03	1.00E + 03
	Avg. Mean	2.78E + 02	8.60E + 02	7.80E + 02	5.95E + 02	7.96E + 02	6.70E + 02	6.45E + 02
	STD.	3.31E + 02	1.84E + 02	2.40E + 02	4.42E + 02	2.26E + 02	4.17E + 02	3.50E + 02
SPSO	Avg. Best	5.34E + 00	2.99E + 01	3.67E + 02	5.66E + 00	2.35E + 01	4.36E + 01	4.80E + 00
	Avg. Worst	9.25E + 02	1.13E + 03	1.04E + 03	1.37E + 03	1.05E + 03	1.66E + 03	1.04E + 03
	Avg. Mean	6.45E + 02	9.37E + 02	8.84E + 02	7.81E + 02	8.95E + 02	9.43E + 02	8.34E + 02
	STD.	2.62E + 02	1.35E + 02	1.36E + 02	3.02E + 02	1.52E + 02	3.25E + 02	2.29E + 02

Table 6
F4: result achieved on peak 10

	Errors	T1	T2	T3	T4	T5	T6	T7
QCSSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	2.53E + 01	5.86E + 02	2.89E + 02	1.26E + 01	5.06E + 02	4.04E + 01	3.25E + 01
	Avg. Mean	1.90E + 00	4.87E + 01	4.45E + 01	1.66E + 00	1.10E + 02	3.22E + 00	5.35E + 00
	STD.	4.91E + 00	1.37E + 02	1.31E + 02	3.60E + 00	1.80E + 02	7.14E + 00	8.38E + 00
QSSA	Avg. Best	1.40E − 12	5.01E − 13	5.12E − 13	8.25E − 13	0.00E + 00	3.38E − 13	0.00E + 00
	Avg. Worst	4.63E + 02	6.68E + 02	6.19E + 02	3.74E + 02	6.08E + 02	7.25E + 02	5.88E + 02
	Avg. Mean	5.82E + 01	3.05E + 02	2.21E + 02	4.29E + 01	3.27E + 02	5.96E + 01	1.01E + 02
	STD.	1.10E + 02	2.71E + 02	2.49E + 02	6.66E + 01	2.25E + 02	1.04E + 02	1.77E + 02
SPSO	Avg. Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Avg. Worst	4.76E + 02	6.13E + 02	6.48E + 02	7.17E + 02	6.72E + 02	6.08E + 02	6.00E + 02
	Avg. Mean	5.93E + 01	2.82E + 02	2.73E + 02	1.18E + 02	3.63E + 02	8.65E + 01	1.16E + 02
	STD.	1.04E + 02	2.62E + 02	2.58E + 02	1.71E + 02	2.49E + 02	1.52E + 02	1.82E + 02

Table 7
F5: result achieved on peak 10

	Errors	T1	T2	T3	T4	T5	T6	T7
QCSSO	Avg. Best	4.09E − 14	4.26E − 14	4.17E − 14	4.09E − 14	4.26E − 14	4.09E − 14	4.09E − 14
	Avg. Worst	1.43E + 01	1.16E + 01	7.89E + 00	6.37E − 07	8.02E + 00	5.45E + 00	1.12E + 01
	Avg. Mean	4.39E − 01	7.06E − 01	5.56E − 01	1.37E − 08	4.99E − 01	2.43E − 01	6.10E − 01
	STD.	2.08E + 00	1.99E + 00	1.66E + 00	9.59E − 08	1.54E + 00	8.86E − 01	1.92E + 00
QSSA	Avg. Best	4.26E − 14	4.17E − 14	4.26E − 14	4.17E − 14	4.26E − 14	4.17E − 14	4.26E − 14
	Avg. Worst	9.17E + 01	8.10E + 01	9.70E + 01	2.22E + 02	5.67E + 01	2.90E + 02	2.74E + 02
	Avg. Mean	3.53E + 01	3.25E + 01	2.65E + 01	4.02E + 01	2.80E + 01	4.38E + 01	3.04E + 01
	STD.	2.62E + 01	2.65E + 01	2.32E + 01	4.10E + 01	1.84E + 01	4.92E + 01	4.29E + 01
SPSO	Avg. Best	4.26E − 14	4.17E − 14	4.26E − 14	4.17E − 14	4.26E − 14	4.09E − 14	4.17E-14
	Avg. Worst	5.94E + 02	5.52E + 02	6.41E + 02	9.55E + 02	6.10E + 01	9.10E + 02	9.36E + 02
	Avg. Mean	8.22E + 01	5.41E + 01	4.49E + 01	8.42E + 01	2.84E + 01	9.88E + 01	8.46E + 01
	STD.	1.33E + 02	8.84E + 01	9.10E + 01	1.43E + 02	1.99E + 01	1.63E + 02	1.71E + 02

Table 8
F6: result achieved on peak 10

	Errors	T1	T2	T3	T4	T5	T6	T7
QCSSO	Avg. Best	4.09E − 14	4.26E − 14	4.17E − 14	4.09E − 14	4.26E − 14	4.09E − 14	4.09E − 14
	Avg. Worst	1.43E + 01	1.16E + 01	7.89E + 00	6.37E − 07	8.02E + 00	5.45E + 00	1.12E + 01
	Avg. Mean	4.39E − 01	7.06E − 01	5.56E − 01	1.37E − 08	4.99E − 01	2.43E − 01	6.10E − 01
	STD.	2.08E + 00	1.99E + 00	1.66E + 00	9.59E − 08	1.54E + 00	8.86E − 01	1.92E + 00
QSSA	Avg. Best	4.26E − 14	4.17E − 14	4.26E − 14	4.17E − 14	4.26E − 14	4.17E − 14	4.26E − 14
	Avg. Worst	9.17E + 01	8.10E + 01	9.70E + 01	2.22E + 02	5.67E + 01	2.90E + 02	2.74E + 02
	Avg. Mean	3.53E + 01	3.25E + 01	2.65E + 01	4.02E + 01	2.80E + 01	4.38E + 01	3.04E + 01
	STD.	2.62E + 01	2.65E + 01	2.32E + 01	4.10E + 01	1.84E + 01	4.92E + 01	4.29E + 01

Table 8 cont.

SPSO	Avg. Best	4.26E − 14	4.17E − 14	4.26E − 14	4.17E − 14	4.26E − 14	4.09E − 14	4.17E − 14
	Avg. Worst	5.94E + 02	5.52E + 02	6.41E + 02	9.55E + 02	6.10E + 01	9.10E + 02	9.36E + 02
	Avg. Mean	8.22E + 01	5.41E + 01	4.49E + 01	8.42E + 01	2.84E + 01	9.88E + 01	8.46E + 01
	STD.	1.33E + 02	8.84E + 01	9.10E + 01	1.43E + 02	1.99E + 01	1.63E + 02	1.71E + 02

Table 9
Performance score of each algorithm on DOPs

		F1 (10)	F1 (50)	F2	F3	F4	F5	F6
QCSSO	T1	0.975645	0.955978	0.831956	0.449703	0.797564	0.844141	0.517966
	T2	0.921049	0.892675	0.603084	0.0885542	0.53388	0.825569	0.603795
	T3	0.881768	0.74245	0.588456	0.171561	0.599233	0.828525	0.685114
	T4	0.97869	0.981183	0.898177	0.618185	0.822885	0.948448	0.622884
	T5	0.930344	0.951612	0.576684	0.182611	0.475045	0.87228	0.69096
	T6	0.895099	0.902026	0.7335	0.213375	0.722497	0.844983	0.548655
	T7	0.886663	0.927386	0.706908	0.30559	0.720747	0.845844	0.610403
CPSO	T1	0.942163	0.940825	0.727937	0.263052	0.687955	0.664818	0.556384
	T2	0.892462	0.887899	0.574953	0.0183243	0.470139	0.611798	0.439927
	T3	0.868731	0.837547	0.580325	0.0375368	0.489701	0.602617	0.431848
	T4	0.976683	0.975418	0.899955	0.276901	0.883281	0.874479	0.630821
	T5	0.889075	0.917639	0.568815	0.0271835	0.462767	0.608712	0.414599
	T6	0.881828	0.873017	0.643585	0.0345534	0.568689	0.538666	0.358982
	T7	0.85684	0.829573	0.65473	0.07386	0.572144	0.588575	0.41351
QSSA	T1	0.776032	0.768278	0.28525	0.22783	0.314371	0.308042	0.285125
	T2	0.733743	0.71041	0.251848	0.0195408	0.165069	0.322602	0.258774
	T3	0.642567	0.718064	0.318484	0.0515741	0.224831	0.359423	0.303288
	T4	0.628939	0.737373	0.233292	0.0949041	0.288899	0.252561	0.227879
	T5	0.737373	0.777055	0.30825	0.0487118	0.17757	0.421606	0.351408
	T6	0.51961	0.520214	0.262468	0.0794769	0.216879	0.243877	0.231668
	T7	0.78629	0.747422	0.322689	0.107792	0.331003	0.36822	0.332159
SPSO	T1	0.875204	0.83808	0.273157	0.0566925	0.345212	0.269387	0.234677
	T2	0.80101	0.784004	0.22755	0.0181545	0.237943	0.336208	0.223524
	T3	0.707618	0.699526	0.289606	0.0221931	0.198411	0.39036	0.368879
	T4	0.604371	0.669144	0.280544	0.0216652	0.250209	0.19413	0.227945
	T5	0.895946	0.877942	0.335961	0.0316889	0.220481	0.487727	0.328684
	T6	0.591665	0.475409	0.169853	0.0140205	0.265328	0.197332	0.215123
	T7	0.726484	0.754176	0.395141	0.0389615	0.365696	0.34295	0.252973

Table 10
Overall final score on DOPs

Algorithm	F1 (10)	F1 (50)	F2	F3	F4	F5	F6	Score [%]
QCSSO	0.092605555	0.09066272	0.112875096	0.046265181	0.106358448	0.137468208	0.097831424	**68.4067**
CPSO	0.0903325	0.0897809	0.106369	0.016963	0.0946551	0.103043	0.0745976	**57.5741**
QSSA	0.06843686	0.069231665	0.044993232	0.014253577	0.038598904	0.051686184	0.045109952	**33.2310**
SPSO	0.07440205	0.072703335	0.04416236	0.004569337	0.042273152	0.050490656	0.042419536	**33.1020**

5.2. Why QCSSO performed better than standard SSA

The searching process of standard SSA is designed for stationary optimization problems in which quick convergence is considered a good feature. The search strategy of standard SSA is not appropriate for dynamic optimization problems where it is necessary to enhance the obtained global optimal solution so far to achieve the foreseen global optima. The original search strategy considers the optimization problem as a single problem instance. For DOPs, it is necessary to consider each dynamic change as a new problem case that must be addressed from scratch. Hence, a better search strategy is required for the DOPs to conform to the dynamic changes, i.e., by forwarding the experience of the optimization process, considering the new environment is somewhat related to the old one.

Also, population diversity is lost in the standard SSA due to expected convergence for the stationary optimization problems. Preserving the population's diversity is crucial for dynamic optimization as the global best changes over time. If the population is collected in a tight region, the individual may not detect changes in the landscape. The obtained solution will not be improved further. From the trajectory path of standard SSA, it can be derived that all the Salp intensely move towards the guided approach by leader Salp in the direction of food position and sometimes converge on local or global optima where food position is located. Because of this feature, the standard SSA is ineffective in discovering and monitoring the evolving global optima in the search landscape.

The QCSSO employs a different mechanism to enhance the performance of standard SSA for dynamic optimization problems. First, a better search strategy is considered to use the experience during the optimization process when a change is detected along with quantum computing to tune the original algorithm for the specific instances of DOPs during the optimization process by taking the actual progress of the search. Second, multi-population with an aging mechanism is applied to discover and monitor the ever-changing global optima in the search landscape. This strategy, in addition to the chaotic logistic operator, helps to control the diversity and convert good local optima into global best during the optimization process in the changing environment.

5.3. Comparative study and the effect of dynamic changes

The performance of QCSSO is measured for all the change types of the dynamic benchmark functions. QCSSO performed better than all its peer algorithms, including CPSO, SPSO, and QSSA. The employed strategy in QCSSO to locate and track global optima; outperformed and indicated its superiority over CPSO, SPSO, and QSSA. The QCSSO achieved excellent results for F1(on both peaks), F2, F5, and F6 test functions of DOPs for all the change types. However, CPSO performed close to QCSSO for chaotic (T4) change types. The hierarchical clustering method and local search strategy enabled CPSO to converge faster and have a better result for the test function F4, chaotic (T4) change types. The regional search strategy helps in searching for optimal solutions in promising sub-regions detected by the clustering method to exploit it effectively. However, the clustering approach is ineffective in generating the sub-swarms consistently, especially in the case of a single particle covering a peak, because there is no improvement during the optimization process in that cluster. From the overall final score in Table 10, the superiority of the QCSSO algorithm can be easily made out.

The algorithm has obtained good results across all change types for functions F1 (on both peaks), F2, and F5. QCSSO has steady achievement on problem F6, i.e., Hybrid composite problem for all seven change types (T1–T7), discovered by the comparative study of best obtained average values from Table 8. Benchmark problem function Rastrigin's (F3) for the change large step (T2) appears as the complex test case for the QCSSO algorithm between all the dynamic optimization benchmark functions.

5.4. Results of the experiment

The overall performance table on DOPs of QCSSO shows the algorithm's superior capability of locating and tracking multiple optima. The algorithm has obtained excellent results across all change types for functions F1 (rotation peak on both peaks), F2 (Composition of Sphere's function), and F5 (Ackley's function) test functions of DOPs for all the change types when it is compared to CPSO, SPSO, and QSSA. Overall, algorithm performance is suitable across all functions. It has performed very well for functions F2, F4, and F5 when comparing the overall score. The F3 function (Composition Rastrigin function) for a large step is the most challenging problem amongst all dynamic benchmark functions of the GDBG framework for the QCSSO algorithm, but still better than the peer algorithms. This analysis is reflected in the overall performance in Table 10, where QCSSO scores highest when compared with other well-regarded algorithms.

6. Conclusion and future work

The current work proposes an extension of the Salp Swarm Algorithm (SSA) with multi-population, quantum computing, and the chaotic logistic map for the dynamic optimization benchmark problems (DOPs) presented in the CEC'2009 special session. For DOPs, an optimization algorithm usually must discover and monitor the multiple optima changing over time. The desirable features of an optimization algorithm include maintaining diversity and multi-population for locating and tracking the optima. In this article, the multi-population is used to discover and monitor the global optima, and quantum computing techniques are used in the equation to increase the searchability of the algorithm. The strategy also helps control the solution's diversity and convert good local optima into global best during the optimization process. Further, a chaotic operator is employed to maintain diversity at the individual level and avoid entrapment in local optima. Implementing chaotic series instead of random in QCSSO is a robust approach to diversify the population, enhance overall performance, and prevent early convergence.

As illustrated in the current contribution, the trial was carried out to evaluate the performance of QCSSO. The proposed algorithm is compared with well-regarded algorithms: QSSA, SPSO, and CPSO. The six dynamic benchmark optimization problems (F1–F6) evaluation results with seven change types (T1–T7) were recorded and analyzed. It shows that QCSSO markedly improves SSA's performance in locating and tracking multiple optima in a dynamic fitness landscape and can find an acceptable solution for most of the DOPs in dynamic and uncertain environments.

Although the currently applied method effectively boosts the performance of SSA for DOPs, a fixed sub-population size is considered in this work, which is not a good approach. Thus, more work can be done to make it self-adaptive based on the optimization progress.

References

[1] Abualigah L., Shehab M., Alshinwan M., Alabool H.: Salp swarm algorithm: a comprehensive survey, *Neural Computing and Applications*, vol. 32, pp. 11195–11215, 2020. doi: 10.1007/s00521-019-04629-4.

[2] Ateya A.A., Muthanna A., Vybornova A., Algarni A.D., Abuarqoub A., Koucheryavy Y., Koucheryavy A.: Chaotic salp swarm algorithm for SDN multi-controller networks, *Engineering Science and Technology, an International Journal*, vol. 22(4), pp. 1001–1012, 2019. doi: 10.1016/j.jestch.2018.12.015.

[3] Branke J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, pp. 1875–1882, 1999. doi: 10.1109/CEC.1999.785502.

[4] Branke J., Kaussler H., Smidt C., Schmeck H.: A Multi-population Approach to Dynamic Optimization Problems. In: I.C. Parmee (ed.), *Evolutionary Design and Manufacture. Selected Papers from ACDM '00*, pp. 299–307, 2000. doi: 10.1007/978-1-4471-0519-0_24.

[5] Bui L.T., Abbass H.A., Branke J.: Multiobjective optimization for dynamic environments. In: *2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2349–2356, 2005. doi: 10.1109/CEC.2005.1554987.

[6] Carlisle A., Dozier G.: Adapting particle swarm optimization to dynamic environments. In: *Proceedings of the International Conference on Artificial Intelligence, Las Vegas, NV, USA*, vol. 1, pp. 429–434, 2000.

[7] Chen Y.R., Tsai C.W., Chiang M.C., Yang C.S.: An Improved Quantum-Inspired Evolutionary Algorithm for Data Clustering. In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3411–3416, 2018. doi: 10.1109/SMC.2018.00578.

[8] Cheng H., Yang S.: Multi-population Genetic Algorithms with Immigrants Scheme for Dynamic Shortest Path Routing Problems in Mobile Ad Hoc Networks. In: *Applications of Evolutionary Computation. EvoApplications 2010,* Lecture Notes in Computer Science, vol. 6024, pp. 562–571, 2010. doi: 10.1007/978-3-642-12239-2_58.

[9] Cobb H.G.: An Investigation Into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments, 1990. NRL Memorandum Report.

[10] Cobb H.G., Grefenstette J.J.: Genetic algorithms for tracking changing environments. In: *Proceedings of the 5th International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA*, pp. 523–530, 1993.

[11] Grefenstette J.J.: Genetic algorithms for changing environments. In: *Parallel Problem Solving from Nature, 2: Proceedings of the Second Conference on Parallel Problem Solving from Nature, Brussels, Belgium, 28–30 September 1992*, pp. 137–144, 1992.

[12] He W., Shi Y.: Multiobjective Construction Optimization Model Based on Quantum Genetic Algorithm, *Advances in Civil Engineering*, vol. 2019, 5153082, 2019. doi: 10.1155/2019/5153082.

[13] Krishnakumar K.: Micro-Genetic Algorithms For Stationary And Non-Stationary Function Optimization. In: *Proceedings of SPIE 1196, Intelligent Control and Adaptive Systems, (1 February 1990)*, pp. 289–296, 1990. doi: 10.1117/12.969927.

[14] Li C., Yang S.: A clustering particle swarm optimizer for dynamic optimization. In: *2009 IEEE Congress on Evolutionary Computation*, pp. 439–446, 2009. doi: 10.1109/CEC.2009.4982979.

[15] Li C., Yang S., Nguyen T.T., Yu E.L., Yao X., Jin Y., Beyer H., Suganthan P.N.: Benchmark Generator for CEC'2009 Competition on Dynamic Optimization, Technical Report, Brunel University London, 2008. http://bura.brunel.ac.uk/handle/2438/5897.

[16] Li S., Yu Y., Sugiyama D., Li Q., Gao S.: A Hybrid Salp Swarm Algorithm With Gravitational Search Mechanism. In: *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 257–261, 2018. doi: 10.1109/CCIS.2018.8691256.

[17] Liu L., Yang S., Wang D.: Particle Swarm Optimization With Composite Particles in Dynamic Environments, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40(6), pp. 1634–1648, 2010. doi: 10.1109/TSMCB.2010.2043527.

[18] Mani A., Patvardhan C.: An adaptative quantum evolutionary algorithm for engineering optimization problems, *International Journal of Computer Applications*, vol. 1(22), pp. 43–48, 2010. doi: 10.5120/444-677.

[19] Manikanta G., Mani A., Singh H.P., Chaturvedi D.K.: Adaptive quantum-inspired evolutionary algorithm for optimizing power losses by dynamic load allocation on distributed generators, *Serbian Journal of Electrical Engineering*, vol. 16(3), pp. 325–357, 2019. doi: 10.2298/SJEE1903325M.

[20] Mirjalili S., Gandomi A.H., Mirjalili S.Z., Saremi S., Faris H., Mirjalili S.M.: Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017. doi: 10.1016/j.advengsoft.2017.07.002.

[21] Nanayakkara T., Watanabe K., Izumi K.: Evolving in dynamic environments through adaptive chaotic mutation. In: *Proceedings of Third International Symposium on Artficial Life and Robotics*, vol. 2, pp. 520–523, 1999.

[22] Nguyen T.T., Yang S., Branke J.: Evolutionary dynamic optimization: A survey of the state of the art, *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, 2012. doi: 10.1016/j.swevo.2012.05.001.

[23] Oppacher F., Wineberg M.: The Shifting Balance Genetic Algorithm: improving the GA in a dynamic environment. In: *GECCO'99: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*, vol. 1, pp. 504–510, 1999.

[24] Parsopoulos K.E., Vrahatis M.N.: Recent approaches to global optimization problems through Particle Swarm Optimization, *Natural Computing*, vol. 1, pp. 235–306, 2002. doi: 10.1023/A:1016568309421.

[25] Pathak S., Mani A., Sharma M., Chatterjee A.: A Novel Salp Swarm Algorithm for Controller Placement Problem. In: N. Kar, A. Saha, S. Deb (eds.), *Trends in Computational Intelligence, Security and Internet of Things. ICCISIoT 2020. Communication in Computer and Information Science*, vol. 1358, pp. 24–36, 2020. doi: 10.1007/978-3-030-66763-4_3.

[26] Pathak S., Mani A., Sharma M., Chatterjee A.: A New Salp Swarm Algorithm for the Numerical Optimization Problems Based on An Elite Opposition-based Learning, *2021 Asian Conference on Innovation in Technology (ASIANCON)*, pp. 1–6, 2021. doi: 10.1109/ASIANCON51346.2021.9544105.

[27] Pathak S., Mani A., Sharma M., Chatterjee A.: A New Quantum-Inspired Salp Swarm Optimization Algorithm for Dynamic Optimization Problem, *2022 IEEE 19th India Council International Conference (INDICON)*, pp. 1–8, 2022. doi: 10.1109/INDICON56171.2022.10040211.

[28] Sayed G.I., Khoriba G., Haggag M.H.: A novel chaotic salp swarm algorithm for global optimization and feature selection, *Applied Intelligence*, vol. 48, pp. 3462–3481, 2018. doi: 10.1007/s10489-018-1158-6.

[29] Sayed G.I., Khoriba G., Haggag M.H.: Hybrid Quantum Salp Swarm Algorithm for Contrast Enhancement of Natural Images, *International Journal of Intelligent Engineering & Systems*, vol. 12, pp. 225–235, 2019. doi: 10.22266/ijies2019.1231.22.

[30] Sun J., Feng B., Xu W.: Particle swarm optimization with particles having quantum behavior. In: *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 1, pp. 325–331, 2004. doi: 10.1109/CEC.2004.1330875.

[31] Sun J., Xu W., Feng B.: A global search strategy of quantum-behaved particle swarm optimization. In: *IEEE Conference on Cybernetics and Intelligent Systems*, vol. 1, pp. 111–116, 2004. doi: 10.1109/ICCIS.2004.1460396.

[32] Vavak F., Jukes K.A., Fogarty T.C.: Performance of a Genetic Algorithm with Variable Local Search Range Relative to Frequency of the Environment Changes. In: *Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22–25, 1998, University of Wisconsin, Madison, Wisconsin*, pp. 602–608, Morgan Kaufmann, San Francisco, CA, 1998.

[33] Xiao B., Wang R., Xu Y., Song W., Wang J., Wu Y.: Salp swarm algorithm based on particle-best. In: *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1383–1387, 2019. doi: 10.1109/ITNEC.2019.8729243.

[34] Xiao B., Wang R., Xu Y., Wang J., Song W., Deng Y.: Simplified Salp Swarm Algorithm. In: *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 226–230, 2019. doi: 10.1109/ICAICA.2019.8873515.

[35] Xing Z., Jia H.: Multilevel Color Image Segmentation Based on GLCM and Improved Salp Swarm Algorithm, *IEEE Access*, vol. 7, pp. 37672–37690, 2019. doi: 10.1109/ACCESS.2019.2904511.

[36] Yang S.: On the Design of Diploid Genetic Algorithms for Problem Optimization in Dynamic Environments. In: *2006 IEEE International Conference on Evolutionary Computation*, pp. 1362–1369, 2006. doi: 10.1109/CEC.2006.1688467.

[37] Yang S., Richter H.: Hyper-learning for population-based incremental learning in dynamic environments. In: *2009 IEEE Congress on Evolutionary Computation*, pp. 682–689, 2009. doi: 10.1109/CEC.2009.4983011.

[38] Yang S., Tinos R.: Hyper-Selection in Dynamic Environments. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3185–3192, 2008. doi: 10.1109/CEC.2008.4631229.

[39] Yang S., Yao X.: Population-Based Incremental Learning With Associative Memory for Dynamic Environments, *IEEE Transactions on Evolutionary Computation*, vol. 12(5), pp. 542–561, 2008. doi: 10.1109/TEVC.2007.913070.

[40] Yazdani D., Akbarzadeh-Totonchi M.R., Nasiri B., Meybodi M.R.: A new artificial fish swarm algorithm for dynamic optimization problems. In: *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, 2012. doi: 10.1109/CEC.2012.6256169.

Affiliations

Sanjai Pathak
Amity University Uttar Pradesh Noida, India, pathak.sanjai@gmail.com

Ashish Mani
Amity University Uttar Pradesh Noida, India, amani@amity.edu

Mayank Sharma
Amity University Uttar Pradesh Noida, India, msharma22@amity.edu

Amlan Chatterjee
California State University Dominguez Hills Carson, CA, USA, achatterjee@csudh.edu

Received: 26.02.2023
Revised: 09.04.2024
Accepted: 09.04.2024

Information for Authors

We accept only the original scientific papers prepared in English. The papers are to be prepared using the LaTeX system. Submitted papers will be refereed by independent reviewers and, if necessary, the Authors may be asked to revise their manuscripts. Proofs will be sent to the Authors for corrections. There is no publication fee. Authors of the accepted papers are eligible to get one hardcopy of the volume containing their contribution free of charge. No postage charges apply.

Our website

https://journals.agh.edu.pl/csci/

Open access

This is an open access journal which means that all content is freely available without charge to the user or his/her institution. This is in accordance with the Budapest Open Access Initiative definition of open access. All printed volumes may be accessed at our website.

Indexing

Google Scholar
`http://scholar.google.com`

Index Copernicus
`http://indexcopernicus.com/`

Directory of Open Access Journals
`http://www.doaj.org`

Open Archives Initiative
`http://www.openarchives.org`

Digital Libraries Federation
`http://fbc.pionier.net.pl/owoc/`

BazTech
`http://baztech.icm.edu.pl`

Worldcat
`http://www.worldcat.org`

WorldWideScience.org
`http://worldwidescience.org`

Sherpa Romeo
`http://www.sherpa.ac.uk/romeo/`

Journal TOCs
`http://www.journaltocs.ac.uk`

SCOPUS
`www.scopus.com`

Web of Science – Emerging Sources Citation Index
`www.webofknowledge.com`

The Computer Science Journal is published by the AGH University of Science and Technology, Krakow Poland. The Editors of the Journal are members of the Faculty of Computer Science, Electronics and Telecommunications and the Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering. The Editorial Board consists of many renowned computer science researchers from all over the world.

The first issue of the Journal was published in 1999. Currently, the Journal is published quarterly, with the main goal to create a forum for exchanging research experience for scientists specialized in different fields of computer science.

Original papers are sought concerning theoretical and applied computer science problems. Example areas of interest are:

- □ theoretical aspects of computer science,
- □ pattern recognition and processing,
- □ evolutionary algorithms,
- □ neural networks,
- □ database systems,
- □ knowledge engineering,
- □ automatic reasoning,
- □ computer networks management,
- □ distributed and grid systems,
- □ multi-agent systems,
- □ multimedia systems and computer graphics,
- □ natural language processing,
- □ soft-computing,
- □ embedded systems,
- □ adaptive algorithms,
- □ simulation.

Abstracts, full versions of the issued volumes and instructions for authors and reviewers may be found at http://csci.agh.edu.pl