2024 International Conference on Emerging Smart Computing and Informatics (ESCI)
AISSMS Institute of Information Technology, Pune, India. Mar 5-7, 2024

Efficiency Comparison of Dataset Generated by LLMs using Machine Learning Algorithms

Premraj Pawade
Dept. of Computer Engineering
AISSMS IOIT
Pune, India
premrajpawade619@gmail.com

Mohit Kulkarni
Dept.of Computer Engineering
AISSMS IOIT
Pune, India
mohit1.kulkarni@gmail.com*

Shreya Naik
Dept. of Computer Engineering
AISSMS IOIT
Pune, India
shreyarnaik03@gmail.com

Aditya Raut
Dept. of Computer Engineering
AISSMS IOIT
Pune, India
adityaraut216@gmail.com

Dr. K. S. Wagh
Dept. of Computer Engineering
AISSMS IOIT
Pune, India
waghks@gmail.com

2024 International Conference on Emerging Smart Computing and Informatics (ESCI) | 979-8-3503-0661-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ESCI59607.2024.10497340

***Abstract*—The constantly expanding field of Large Language Models (LLMs) offers exciting opportunities for various domains. These powerful models, such as GPT-3.5, Bard, and Bing, can produce massive amounts of text-based data, creating new avenues for generating synthetic datasets. The primary focus of this research is to explore the effectiveness of LLMs in creating high-quality, structured datasets for different ML applications. Specifically, this study concentrates on password strength prediction. It compares the performance of three prominent LLMs - Bard, ChatGPT, and BingAI - in generating datasets of text-based passwords with their corresponding strength levels. This research uses a diverse set of ML models, including traditional algorithms like XGBoost, Random Forest, etc., to evaluate the generated datasets. The evaluation process assesses their performance, generalization, and adaptability. This research contributes to the growing field of LLM-based data generation by demonstrating their effectiveness in creating valuable datasets for specific machine learning applications. The findings of this study pave the way for further exploration of LLMs' capabilities for diverse data types and tasks, potentially unlocking new avenues for advancements in various machine learning domains.**

***Index Terms*—Large Language Models (LLMs), Machine Learning, Artificial Data Generation, password-strength, Generative AI, Evaluation Metrics, Data Augmentation.**

I. Introduction

Large Language Models (LLMs) have revolutionized the field of natural language processing (NLP) and machine learning (ML). Models like Bard, ChatGPT, and BingAI showcase exceptional proficiency in generating vast amounts of text-based data, offering new possibilities for data augmentation and mitigating challenges associated with data scarcity. In ML applications, the insatiable demand for data is indisputable. However, collecting and preparing high-quality datasets is often time-consuming and laborious, particularly in domains where data is scarce or challenging. Statistical reports indicate that the efforts involved in traditional data collection methods often consume a substantial portion of the overall project timeline. In fact, on average, data collection and preprocessing can account for up to 80% of the time invested in the early stages of ML projects. Moreover, this process becomes particularly arduous in domains where data is scarce or challenging to obtain, amplifying the need for innovative solutions. Recent strides in LLM research highlight the potential of these models to generate artificial datasets comparable to real-world counterparts in quality and utility. This study delves into the efficacy of LLMs in generating password-strength datasets, centring on Bard, ChatGPT, and BingAI. To rigorously assess the quality of the generated datasets, we employ established machine learning models, including XGBoost, Random Forest, Decision Tree, SVM, and KNN classifiers. These models undergo initial training on a benchmark dataset (000WebHost) to establish baseline performance in password strength prediction. Subsequently, the XGBoost classifier, identified as the most effective model, scrutinizes the generated datasets. This detailed evaluation provides valuable insights into their effectiveness compared to the benchmark.

This research provides important insights into the potential of Language Model Models (LLMs) to create high-quality datasets for password strength prediction and various other Machine Learning (ML) applications. By overcoming the limitations of traditional data collection methods, LLMs enable ML practitioners to develop more accurate and reliable models, ushering in a new era of excellence in machine learning. The study also examines the impact of dataset characteristics, data preprocessing methods, and the role of LLMs in data enhancement, providing a nuanced perspective on the critical components that shape the dataset generation process and subsequent model evaluation. [1]

II. Literature Survey

Giuseppe Destefanis et al., assesses the effectiveness of two sophisticated AI models, namely GPT-3.5 and Bard, in

generating Java code based on a given description of the function. A thorough analysis of the accuracy of the Java code generated by both models, utilizing test cases, was provided. [2]

Ryunosuke Noda et al., has studied ChatGPT and Bard models and their possible applications in nephrology. The development of large language models (LLMs) trained on extensive data has played a significant role in recent developments in AI. GPT-4 has shown high performance levels in general medical exams but its performance is yet to be determined. [3]

Afgiansyah has investigated the neutrality of AI-based tools in information dissemination, specifically Microsoft's Bing Chat powered by GPT and Google's Bard, across three geopolitical topics. The study used the Gamson and Modigliani framing model to evaluate that although there were attempts to maintain neutrality, subtle Western perspectives were evident in the narratives, particularly American ones. [4]

Md.Saidur Rahamanet et al., has conducted research on two leading AI contenders, Google's Bard with LaMDA and Open AI's ChatGPT. LaMDA is a neural language model based on the transformer architecture, which is already trained on the data of online chat. ChatGPT is built on the GPT-3.5 model and includes a reinforcement learning technique with human feedback. [5]

Indira Mannuela et al., in their research has worked on password security and factors that make passwords vulnerable. These include the password length, the elements used in the password, the reuse of passwords, and frequently altering passwords. Considering these criteria when creating a password is essential to ensure security. [6]

Pan Singh Dhoni et al., in their research focuses on generative AI and its implications for cybersecurity. Technology offers numerous benefits, but it may also threaten personal privacy. While the advent of Generative AI promises a bright future, it also presents challenges, primarily cybersecurity. [7]

Gongzhu Hu in his paper reviews different metrics for measuring password quality and comparing their strengths and weaknesses by conducting experiments to crack passwords of varying strengths. The primary concern with passwords is their strength, which refers to how easily they can be assumed by someone who wishes to access a resource by pretending to be you. [8]

Ding Wang et al., conducted research on random-forest-based guessing models. Through experiments using 13 large real-world password datasets, RFGuess was found effective for trawling guessing scenarios and has comparable assuming success rates to other leading models. [9]

Umar Farooq proposed a model to combat online and offline cyber attacks that suggests implementing multiple machine-learning algorithms to force users to choose a strong password. This provides a common and efficient way to defend against these attacks. [10]

Vijaya MS et al., has utilized machine-learning techniques for predicting password strength which was approached as a classification problem. The commonly used supervised machine learning algorithms were used to train the model. The results were compared, and it was observed that the Support Vector Machine algorithm performed better than the others. The existing password strength-checking tools were used to compare the models. [11]

Fangyi Yu has compared deep learning-based password-guessing techniques that do not require prior knowledge about password structures or user behaviour. The models used for comparison are Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), autoencoders, and attention mechanisms. The proposed design uses variations of the Improved Training of Wasserstein GANs (IWGANs) for password guessing under non-targeted offline attacks. [12]

Dario Pasquini et al., presents a new approach for password guessing called deep generative model representation learning. The research demonstrates that using an abstract password representation can provide valuable and flexible properties that can open new avenues in the field of password guessing, which is currently an active area of research. [13]

III. Overview of LLM Models

- Bard was trained on a dataset that includes a variety of text formats, such as news articles, blog posts, and social media posts, via a technique called masked language modeling (MLM). In MLM, the model is given a sentence with some words replaced with masks, and then it tries to predict the missing words. This helps the model to learn the relationships between words in a sentence

- Bing Chat was trained on web text data, including Common Crawl, Wikipedia, news articles, social media posts, and other publicly available text, via a technique called denoising autoencoder (DAE). In DAE, the model is given a noisy version of a sentence, and then it tries to reconstruct the original sentence. This helps the model to learn the structure of human language.

- GPT-3.5 was trained on WebText2 or OpenWebText2 (22 percent), a massive dataset ranging between 66GB and 1.5 TB. It was trained via a technique called reinforcement learning, where the model is given a reward signal for generating text that is similar to human-written text. This helps the model to learn to generate text that is fluent and natural.

IV. Design Methodology

A. Overview

We have developed a system that evaluates and compares the quality of datasets generated by three different LLM models: Bard, Bing, and GPT-3.5. We used well-established machine learning models, including XGBoost, Random Forest, Decision Tree, SVM, and KNN classifiers, to train them on a benchmark dataset (000WebHost) for predicting password strength. Before training, the Webhost dataset was preprocessed and tokenized. Among the models, XGBoost showed the highest accuracy, precision, recall, and F1 score, making it the most promising model. We also generated datasets from all three LLM models using a base prompt and prompt engineering for iterative improvements. The generated datasets underwent refinement and preprocessing for the subsequent testing phase, serving as test data for the previously trained model. Finally, we assessed the results and compared them to the benchmark to determine the effectiveness of LLM Models in generating proficient datasets.

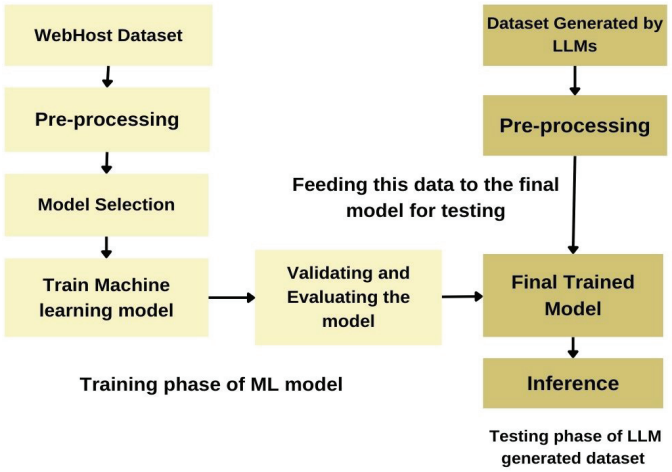

Fig. 1. System Architecture

B. Data Acquisition

*1) **Webhost**:* In 2015, a data breach exposed sensitive information from 000Webhost, a free PHP and MySQL web space provider. The leaked dataset included usernames, plaintext passwords, email addresses, IP addresses, and names. Notably, passwords adhered to a composition policy requiring a minimum of 6 characters with a combination of letters and numbers. The breach, attributed to a hacker exploiting a bug in an outdated PHP version, compromised 669,643 unique passwords. The dataset featured two columns: plaintext passwords and strength values (0 to 2) indicating weak, medium, or strong. A subsequent analysis involved creating a strength parameter using Georgia Tech University's PARS tool, integrating various commercial password meters. The dataset was refined to 0.7 million passwords, aligning with classifications from multiple algorithms for enhanced accuracy.

*2) **Preprocessing**:* During initial data preprocessing, the WebHost passwords dataset underwent crucial cleaning steps, ensuring data integrity and null value eradication using the 'dropna' method. Rigorous quality checks eliminated duplicates, preserving dataset cleanliness. A structured format was created, maintaining data consistency and categorical value uniqueness for subsequent analyses. To prepare the textual password data for machine learning, a word_divider function broke passwords into individual characters, revealing intricate patterns. Employing Scikit-learn's TfidfVectorizer facilitated feature extraction, utilizing the word_divider function as a tokenizer. Utilizing character n-grams (e.g., (2, 4)) via analyzer='char' increased pattern inclusion. Also, it was divided into a 75%-25% split for training and testing, respectively. This multi-step pre-processing, handling null values, character-level tokenization, and using character n-grams, forms a sturdy foundation for subsequent machine learning model training.

C. Model Training and Selection

*1) **Model Training**:* After processing the password dataset, a thorough investigation was conducted to determine the best classification model for accurately predicting password strength. As password strength needs to be categorized into specific classes like 0, 1 and 2, we evaluated various supervised machine learning models that could perform robust classification tasks. Five distinct models were meticulously considered: Random Forest, XGBoost, Decision Tree, SVM Classifier, and KNN Classifier. Each model underwent rigorous training and assessment to ensure its generalizability and ability to accurately classify passwords into distinct strength levels across diverse password characteristics. Through this rigorous evaluation process identified the model demonstrating superior classification performance and possessing the most significant potential for real-world applications.

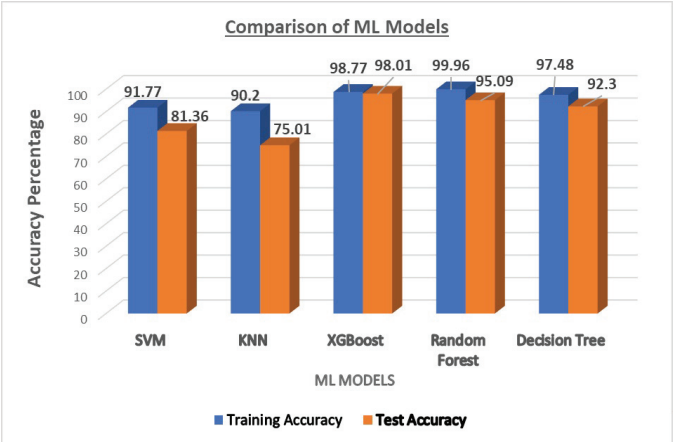

Fig. 2. Comparison of ML Algorithms

TABLE I
CLASSIFICATION MATRIX OF ML MODELS

ML Model	Accuracy	Precision	Recall	f1score
XGBoost	**98.01**	**0.98**	**0.96**	**0.97**
SVM	81.36	0.75	0.56 3	0.56
KNN	75.01	0.63	0.48	0.52
Random Forest	95.09	0.95	0.90	0.95
Decision Tree	92.3	0.88	0.88	0.88

*2) **Decision to Favor XGBoost**:* Our decision to favour XGBoost was based on the following factors:

- **Exceptional performance**: As shown in Table 1, XGBoost outperformed other models in all evaluation metrics.
- **Ensemble learning:** XGBoost's capability of combining the predictions of several weak learners resulted in a more accurate and robust overall model.
- **Scalability and efficiency:** XGBoost's inherent properties make it ideal for deployment in real-world scenarios where efficiency and scalability are critical.

D. Algorithms Applied

***XGBoost Working**:* Among the diverse machine learning models explored for password strength prediction, XGBoost (Extreme Gradient Boosting) emerged as the most effective. This section delves deeper into its workings, highlighting its key strengths and differentiating it from the other models considered. At its core, XGBoost leverages an ensemble learning approach, sequentially adding weak learners (typically decision trees) to progressively refine the prediction of password strength. Each new learner focuses on correcting the errors of its predecessors, resulting in a more accurate and robust model. This iterative learning process, known as gradient boosting, allows XGBoost to effectively capture complex relationships within the data and overcome the limitations of individual weak learners.

***XGBoost Classifier - Comprehensive Explanation**:* XGBoost (eXtreme Gradient Boosting) is a versatile and powerful ensemble learning algorithm widely used for binary classification. The goal of the algorithm is to minimize the objective function consisting of a loss term and a regularization term for each tree in the ensemble.

a) Input:: Given a training set $\{(u_i, v_i)\}_{i=1}^{N}$, a differentiable loss function $L(v, F(u))$, M represents number of weak learners, α is the learning rate.

- Set a constant value as the model's initialization:

$$\hat{f}_{(0)}(u) = \arg\min_{\theta} \sum_{i=1}^{N} L(v_i, \theta).$$

- For $m = 1$ to M:

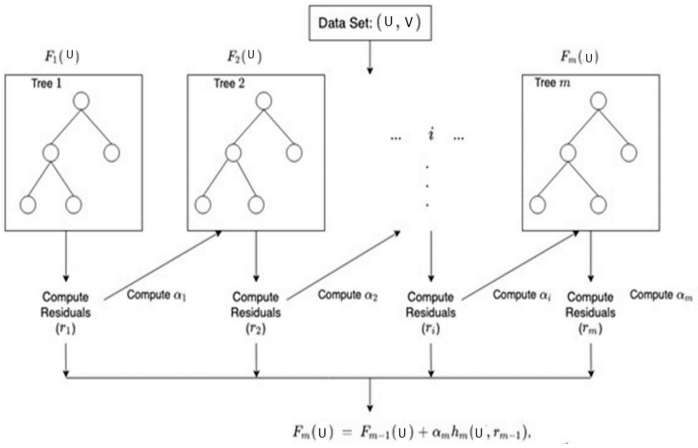

Fig. 3. Working of XGBoost algorithm

1) Calculate the gradients and Hessians:

$$\hat{g}_m(u_i) = \left[\frac{\partial L(v_i, f(u_i))}{\partial f(u_i)}\right]_{f(u)=\hat{f}_{(m-1)}(u)}$$
$$\hat{h}_m(u_i) = \left[\frac{\partial^2 L(v_i, f(u_i))}{\partial f(u_i)^2}\right]_{f(u)=\hat{f}_{(m-1)}(u)}$$

2) Fit a base learner (e.g., tree) using the training set:

$$\left\{u_i, -\frac{\hat{g}_m(u_i)}{\hat{h}_m(u_i)}\right\}_{i=1}^{N}$$

by solving the optimization problem:

$$\hat{\phi}_m = \arg\min_{\phi\in\Phi} \sum_{i=1}^{N} \frac{1}{2}\hat{h}_m(u_i)\left[\phi(u_i) - \frac{\hat{g}_m(u_i)}{\hat{h}_m(u_i)}\right]^2$$

3) Update the model:

$$\hat{f}_{(m)}(u) = \hat{f}_{(m-1)}(u) + \alpha\hat{\phi}_m(u)$$

- The total of all the trees' predictions yields the final model prediction:

$$\hat{v}(u) = \hat{f}_{(M)}(u) = \sum_{m=1}^{M} \hat{f}_m(u)$$

This ensemble approach, consisting of sequentially added trees, optimizes the overall objective function by minimizing the loss and penalizing complex models through regularization.

- **Hyperparameters** Key hyperparameters include the learning rate (α), the number of weak learners (M), and any other relevant hyperparameters. Tuning these parameters is crucial for achieving optimal model performance.
- **Objective Function Components** The loss term $L(v_i, \hat{v}_i)$ represents the discrepancy between the true label v_i and the predicted probability $\hat{v}_i$. Depending on the problem,

this could be the logistic loss, squared error loss, or another suitable loss function.
- **Stopping Criteria** XGBoost often employs stopping criteria during the training process to prevent overfitting. Early stopping, for example, evaluates the model's functioning on data set validation and stops training when improvements stagnate.
- **Final Model Interpretation** The final model prediction $\hat{v}(u)$ is obtained by summing the predictions from all the trees. This ensemble approach produces a robust and accurate classifier.

Key Differentiators: Several key factors set XGBoost apart from the other models considered:

- **Regularization:** XGBoost incorporates built-in techniques like L1 and L2 regularization to prevent overfitting. This ensures the model's generalizability and accurate performance across diverse password datasets.
- **Scalability:** XGBoost's efficient and parallel learning capabilities make it ideal for handling large datasets, which is a critical requirement for real-world password-strength prediction applications.
- **Gradient Boosting:** This core technique enables XGBoost to learn from the errors of previous learners, leading to a more efficient learning process compared to other models like Random Forest, Decision Tree, SVM Classifier, and KNN Classifier.

V. Observations and Results

A. *Dataset Generation*

*1) **GPT-3.5**:* During the testing phase, GPT-3.5 demonstrated exceptional speed in understanding user requirements and delivering requested password sets promptly. It is noteworthy that when asked for 200 passwords, the model created an average of 125 passwords, indicating an efficiency rate of approximately 62.5%. Interestingly, when requested for 100 passwords, GPT-3.5 produced an almost accurate count of 100. The model frequently generated passwords by making slight variations or modifications from the preceding ones, usually following an evident pattern of appending or modifying digits at the end of the password. Additionally, GPT-3.5 occasionally produced exceptionally long passwords, indicating a preference for lengthy passwords and displaying a bias towards strength classification of 2. Remarkably, GPT-3.5 exhibited fewer instances of hallucinations compared to other models, indicating its reliability in generating accurate and secure passwords.

*2) **Bard**:* Bard was compared with GPT-3.5 and Bing for generating password datasets. Although it was slower than GPT-3.5 in understanding user needs, it outperformed Bing in doing so. However, Bard faced difficulties in generating larger requests of 200 passwords and often refused or produced unrealistic passwords. Like GPT-3.5, Bard generated slightly modified passwords. Also, it showed a balanced strength distribution. It added complexity by including proverbs and phrases in longer passwords. However, prolonged interactions with Bard led to escalating hallucinations, which made it unreliable and unstable for generating passwords under sustained pressure. This flaw poses a significant challenge to Bard's long-term suitability for password generation.

*3) **Bing**:* We found that Bing's response for generating passwords was not up to the mark when compared to other models. It consistently gave an average of 15-20 passwords per prompt, which was lower than what we expected. This meant that we had to repeat the original prompt multiple times during each chat session, which affected the efficacy of generating passwords. Additionally, obtaining unique passwords using Bing was a challenge. We had to create additional prompts requesting 50 new and distinct passwords within the specified format, but even then, we were unable to get consistent results. Sometimes we received a decent number of unique passwords, but other times, we only got 20-30 passwords. Bing's performance was highly irregular, making it unreliable for large-scale generation tasks. Although there were some instances of exceptional performance, the overall inconsistency and inadequate output made it less suitable than other models.

B. *Limitations faced during Dataset Generation*

The limitations encountered during this research have highlighted some crucial aspects that affect the quality of the dataset generated. Although Large Language Models (LLMs) have remarkable capabilities, they faced challenges in exploring the diverse range of potential passwords. Sometimes, the inclination to replicate familiar patterns led to the generation of predictable outputs that compromised the dataset's efficiency. The subjective nature of password strength evaluation has emerged as a significant hurdle since users have diverse priorities and preferences. The trade-off between creativity and practicality became apparent since too much creativity might enhance security, but it often resulted in impractical, complex passwords, which undermined usability.

The dependence of LLMs on the quality and diversity of their original training data became evident. Biases or prevalent patterns within the training data could influence the generated passwords, potentially affecting their utility in practical applications. It is also crucial to acknowledge that LLMs considered the semantic meaning of passwords, a factor often overlooked by the trained ML models, resulting in eventual reduced accuracy.

C. *Result*

Among the tested LLMs, Bing demonstrated the highest accuracy at 60.45%, followed by GPT-3.5 (57.53 %) and Bard (53.06%), respectively. The accuracy was assessed through a comparison with the benchmark dataset, 000WebHost, utilizing the trained XGBoost classifier model. Additionally, observations were made during the generation of datasets by

the LLMs, taking into account five parameters as specified in Table II.

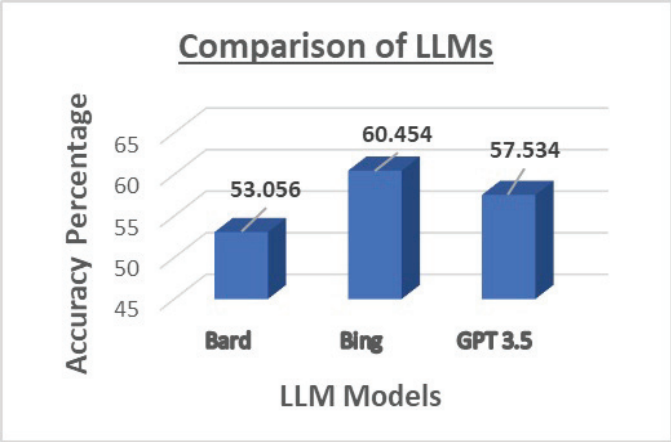

Fig. 4. Comparison of Large Language Models

TABLE II
COMPARISON OF DATASETS GENERATED

Parameters	Bard	GPT-3.5	Bing
Avg len of passwords generated	14.43	14.62	13.07
Avg No. of passwords generated per prompt	90.73	125.6	20.56
Hallucinations	Moderate	Low	High
Standard Deviation of strength distribution	122.47	540.42	312.62
Bias towards strength	1	2	2

VI. CONCLUSION

This study explores how Large Language Models (LLMs) can be used to create high-quality datasets, with a specific focus on predicting password strength. After conducting a thorough examination, the study highlights the potential and limitations of using LLMs for dataset creation.The experimentation results show that XG Boost is the most effective, achieving an impressive 98.01% accuracy rate in password strength prediction. Among the LLMs tested, Bing LLM stands out with an accuracy rate of 60.45%, outperforming Bard and ChatGPT.The study suggests a new approach that leverages the strengths of LLMs while acknowledging their limitations. It encourages further research to enhance LLM capabilities, improve training data diversity, and develop methodologies that balance security and usability concerns.

Ultimately, the study demonstrates the effectiveness of LLMs in dataset creation and their potential to accelerate machine learning advancements. LLMs offer novel approaches for generating accurate and reliable models across diverse applications, promising a new era in machine learning.

ACKNOWLEDGMENT

This report's success and outcome demanded a great deal of direction. We are thankful to our guide, Dr. K. S. Wagh, for his experience and support. We appreciate him for his insight and information, which were extremely useful throughout the course of the study. We would also like to thank Dr. S.N. Zaware, Head of the Computer Engineering Department, AISSMS Institute of Information Technology, for her assistance. We would also like to thank Principal Dr.P.B. Mane for his energetic and helpful advice during the project, as well as for providing the essential facilities that enabled us to complete our dissertation work.

REFERENCES

[1] H. Sarih, A. P. Tchangani, K. Medjaher, and E. Péré, "Data preparation and preprocessing for broadcast systems monitoring in phm framework," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1444–1449, IEEE, 2019.

[2] G. Destefanis, S. Bartolucci, and M. Ortu, "A preliminary analysis on the code generation capabilities of gpt-3.5 and bard ai models for java functions," *arXiv preprint arXiv:2305.09402*, 2023.

[3] R. Noda, Y. Izaki, F. Kitano, J. Komatsu, D. Ichikawa, and Y. Shibagaki, "Performance of chatgpt and bard in self-assessment questions for nephrology board renewal," *medRxiv*, pp. 2023–06, 2023.

[4] A. Afgiansyah, "Artificial intelligence neutrality: Framing analysis of gpt powered-bing chat and google bard," *Jurnal Riset Komunikasi*, vol. 6, no. 2, pp. 179–193, 2023.

[5] M. S. Rahaman, M. Ahsan, N. Anjum, M. M. Rahman, and M. N. Rahman, "The ai race is on! google's bard and openai's chatgpt head to head: an opinion article," *Mizanur and Rahman, Md Nafizur, The AI Race is on*, 2023.

[6] I. Mannuela, J. Putri, M. S. Anggreainy, *et al.*, "Level of password vulnerability," in *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, vol. 1, pp. 351–354, IEEE, 2021.

[7] P. Dhoni and R. Kumar, "Synergizing generative ai and cybersecurity: Roles of generative ai entities, companies, agencies, and government in enhancing cybersecurity," *Authorea Preprints*, 2023.

[8] G. Hu, "On password strength: a survey and analysis," *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 165–186, 2018.

[9] D. Wang, Y. Zou, Z. Zhang, and K. Xiu, "Password guessing using random forest," in *Proc. USENIX SEC 2023*, pp. 1–18, 2023.

[10] U. Farooq, "Real time password strength analysis on a web application using multiple machine learning approaches," *International Journal of Engineering Research and Technology*, vol. 9, no. 12, pp. 359–364, 2020.

[11] M. Vijaya, K. Jamuna, and S. Karpagavalli, "Password strength prediction using supervised machine learning techniques," in *2009 international conference on advances in computing, control, and telecommunication technologies*, pp. 401–405, IEEE, 2009.

[12] F. Yu, "On deep learning in password guessing, a survey," *arXiv preprint arXiv:2208.10413*, 2022.

[13] D. Pasquini, A. Gangwal, G. Ateniese, M. Bernaschi, and M. Conti, "Improving password guessing via representation learning," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1382–1399, IEEE, 2021.