Embedding with Large Language Models for Classification of HIPAA Safeguard Compliance Rules

Md Abdur Rahman
Dept. of Intelligent Systems and Robotics
University of West Florida
FL, United States
E-mail: mr252@students.uwf.edu

Md Abdul Barek
Dept. Intelligent Systems and Robotics
University of West Florida
FL, United States
E-mail: mb381@students.uwf.edu

ABM Kamrul Islam Riad
Dept. of Intelligent Systems and Robotics
University of West Florida
FL, United States
E-mail: ai62@students.uwf.edu

Md Mostafizur Rahman
Cybersecurity and Information Technology
University of West Florida
FL, United States
E-mail: mr240@students.uwf.edu

Md Bajlur Rashid
Cybersecurity and Information Technology
University of West Florida
FL, United States
E-mail: mr248@students.uwf.edu

Md Raihan Mia
Dept. of Computer Science
Marquette University
WI, United States
E-mail: mdraihan.mia@marquette.edu

Hossain Shahriar
Center for Cybersecurity
University of West Florida
Pensacola, FL, USA
E-mail: hshahriar@uwf.edu

Guillermo Francia III
Center for Cybersecurity
University of West Florida
Pensacola, FL, USA
E-mail: gfranciaiii@uwf.edu

Fan Wu
Department of Computer Science
Tuskegee University
Tuskegee, AL, United States
E-mail: fwu@tuskegee.edu

Alfredo Cuzzocrea
iDEA Lab
University of Calabria
Rende, Italy
E-mail: alfredo.cuzzocrea@unical.it

Sheikh Iqbal Ahamed
Department of Computer Science
Marquette University
WI, United States
E-mail: sheikh.ahamed@marquette.edu

***Abstract*—Although software developers of mHealth apps are responsible for protecting patient data and adhering to strict privacy and security requirements, many of them lack awareness of HIPAA regulations and struggle to distinguish between HIPAA rules categories. Therefore, providing guidance of HIPAA rules patterns classification is essential for developing secured applications for Google Play Store. In this work, we identified the limitations of traditional Word2Vec embeddings in processing code patterns. To address this, we adopt multilingual BERT (Bidirectional Encoder Representations from Transformers) which offers contextualized embeddings to the attributes of dataset to overcome the issues. Therefore, we applied this BERT to our dataset for embedding code patterns and then uses these embedded code to various machine learning approaches. Our results demonstrate that the models significantly enhances classification performance, with Logistic Regression achieving a remarkable accuracy of 99.95%. Additionally, we obtained high accuracy from Support Vector Machine (99.79%), Random Forest (99.73%), and Naive Bayes (95.93%), outperforming existing approaches. This work underscores the effectiveness and showcases its potential for secure application development.**

***Index Terms*—HIPAA Compliance, HIPAA Technical Safeguard Rules, Large Language Models, BERT for code processing;**

I. Introduction

As Android devices become increasingly popular due to their flexibility and affordability, the need for secure software development remains critical. The Android platform is a main target for attackers seeking to exploit security vulnerabilities within mobile applications [1]. Therefore, software developers need to ensure their apps must be HIPAA (Health Insurance Portability and Accountability Act) compliant because mHealth apps handle sensitive personal health information (PHI). The Open Web Application Security Project (OWASP) has stressed the significance of tackling these vulnerabilities through its OWASP Mobile Top 10 list [2]. This list identifies the most prevalent security issues in mobile apps and serves as a key benchmark for evaluating tools designed to detect vulnerabilities [3]. Following OWASP guidelines is essential for improving app security and protecting users.

The number of reported software vulnerabilities has steadily increased over the past decade (Fig. 1). Fig. 2 shows vulnerabilities distribution reported over the years, grouped by severity levels. It highlights how vulnerabilities have varied in

frequency and severity across different time periods.

The integration of large language models (LLMs) into the detection of Android code vulnerabilities presents a promising and innovative approach to improving code analysis. This method not only enhances the accuracy of vulnerability detection but also addresses some of the limitations associated with traditional detection techniques. A growing method for detecting vulnerabilities in Android code involves utilizing LLMs for code analysis. The application of LLMs has a track in 2010s with the development of Word2Vec [4]. In fact, a shallow neural network was designed to learn word embeddings—dense vector representations of words—from extensive datasets. This helped models understand semantic relationships between words, which can also be applied to programming code.

Bidirectional Encoder Representations from Transformers (BERT) was innovated by Google which is an advancement over Word2Vec [5]. BERT's bidirectional training enabled it to learn context from both preceding and following text, and it provides acute understanding of language. Today, LLMs like BERT are essential for detecting and addressing security vulnerabilities in Android applications.

The use of LLMs for code analysis began around 2017, with early applications focused on code completion. In November 2019, GPT-2 [6] was introduced, trained on a vast dataset that included source code. These models, based on their understanding of code structure and context, were able to predict the next logical piece of code following a given input.

In 2020, OpenAI [7] launched GPT-3 [8], utilizing 175 billion parameters, significantly enhancing its ability to generate human-like text and code from task descriptions. Recent studies [9,10] have highlighted LLMs' capability to comprehend and analyze code. However, as of now, there is no thorough comparison in the literature of LLMs' effectiveness in detecting Android code vulnerabilities.

Akter et al. and Riad et al. focus on evaluating and enhancing HIPAA compliance in mHealth apps and AI-driven health devices. Akter et al. specifically assess Laravel-based mHealth applications for technical compliance, while Riad et al. explore security and privacy measures in AI-powered mHealth devices [12-13]. Both studies contribute valuable insights into ensuring HIPAA adherence in modern healthcare technologies, highlighting the importance of data protection and privacy in health applications.

Several researchers have utilized machine learning (ML), big data, and quantum machine learning (QML) for detecting vulnerabilities and security threats. Rahman (2020) applied ML algorithms to detect Distributed Denial of Service (DDoS) attacks, contributing to network security advancements [14-19]. Akter et al. (2023) applied deep learning for prediction of cryptocurrency price declines at early stage, while Rahman and Hossain explored clustering-based intrusion detection using Hadoop-PySpark in big data environments. Additionally, Rahman et al. (2023) developed Quantum Generative Adversarial Networks (QGAN) for intrusion detection, and fine-tuned quantum classifiers for cyber-attack detection, highlighting the role of QML in enhancing cybersecurity. These works underscore the growing importance of advanced ML and QML techniques in vulnerability detection and cybersecurity.

Recent research has increasingly focused on using Large Language Models (LLMs) for detecting HIPAA rule violations and vulnerabilities [20]. Rahman et al. (2024) applied pre-trained multilingual BERT embeddings to enhance the detection of malicious prompt injection attacks [21]. Recent studies by Rahman et al. (2025) explore advanced AI methods to enhance cybersecurity: one leverages GANs for synthetic data to boost intrusion detection [31], while the other shows how LLMs can reduce reliance on large training datasets [32]. Additionally, Barek et al. (2024) explored to mitigate outputs which is insecured. Akter et al. (2024) introduced an authentic learning approach to address data poisoning vulnerabilities in LLMs [22]. These works highlight the potential of LLMs in identifying security risks and ensuring compliance with HIPAA regulations, offering new approaches for safeguarding sensitive health data.

The works of Cuzzocrea et al. address critical aspects of data management and privacy. They explore efficient data stream processing and robust frameworks for privacy-preserving OLAP, demonstrating techniques for optimizing query responses and advanced visualization in multidimensional data cubes. These studies underscore the importance of effective data handling and security, relevant for HIPAA compliance in health applications.

According to this, we selected multilingual BERT from Hugging Face that were already trained specifically for code vulnerability analysis:

- We used it to embed HIPAA code patterns as it captures contextual meanings of code through bidirectional analysis, while Word2Vec only provides static embeddings.
- We processed dataset attributes which improves performance by transforming raw, unstructured data into clean, organized, and analyzable formats and concatenating the attributes.

In this work, Section II discusses HIPAA-related concepts. Sections III and IV cover LLMs for software vulnerabilities and BERT applications, respectively. The dataset, methods, and results are presented in the following three sections. Finally, we summarize our findings.

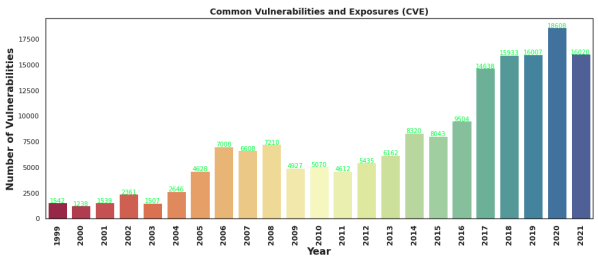

Fig. 1. This bar diagram shows the number of vulnerabilities reported by year.

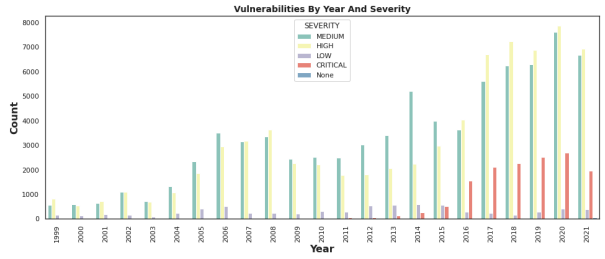

Fig. 2. This illustrates the distribution of vulnerabilities over the years, categorized by severity.

II. HIPAA

A. HIPAA Compliance

HIPAA compliance for Android applications is crucial for developers building mobile health (mHealth) apps that handle sensitive patient data. Health Insurance Portability and Accountability Act (HIPAA) creates stringent guidelines inoder to protect personal health information (PHI), ensuring that apps adhere to privacy, security, and confidentiality standards.

Android apps must implement secure data storage, transmission, and processing mechanisms. This includes encryption of PHI both in transit and at rest, as well as secure authentication and access controls to prevent unauthorized access. Developers should also implement secure logging and auditing features to track access and changes to sensitive data.

Moreover, Android apps need to comply with HIPAA's breach notification rules, requiring developers to inform users and authorities in case of data breaches. Failing to meet these standards can result in severe penalties. Therefore, ensuring HIPAA compliance is essential for building trustworthy and secure Android-based healthcare applications, protecting patient privacy, and avoiding legal risks.

B. HIPAA Technical Safeguard Rules

HIPAA's Technical Safeguard Rules play a critical role in protecting electronic protected health information (ePHI). These rules, outlined by the Health Insurance Portability and Accountability Act (HIPAA), require healthcare providers and mHealth app developers to implement stringent security measures to safeguard patient data. These measures are intended to prevent unauthorized access, alteration, or exposure of sensitive health information. By complying with these regulations, organizations can protect patient data from potential security breaches and cyberattacks. Key defenses, such as encryption, access controls, and activity monitoring systems, must be employed to ensure the confidentiality and integrity of ePHI. Non-compliance with these security standards can lead to serious legal consequences and privacy risks, making it imperative for any entity handling ePHI to fully adhere to HIPAA's technical and administrative safeguards.

Key components of the Technical Safeguard Rules include access control, which ensures only authorized individuals can access ePHI. This is typically achieved through mechanisms like unique user identification, automatic logoff, and encryption. Audit controls must also be implemented to record and examine activity related to ePHI, allowing organizations to track data access and prevent breaches.

Integrity controls ensure that ePHI is not improperly altered or destroyed, maintaining its accuracy and reliability. Finally, transmission security safeguards require encryption methods to protect ePHI when being transmitted over electronic networks. By adhering to these rules, organizations can maintain the confidentiality, integrity, and security of sensitive patient information.

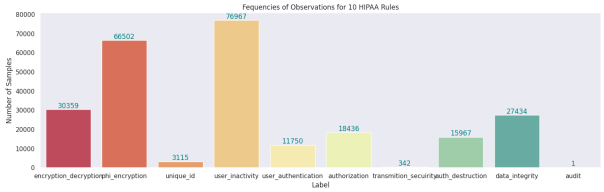

Fig. 3. This diagram shows the frequencies of observations based on 10 HIPAA rules.

III. Large Language Models for Software Vulnerabilities

Large Language Models (LLMs) have demonstrated potential for identifying software vulnerabilities, but they come with notable limitations. Their accuracy averages around 60% across datasets, and while they excel at detecting simpler vulnerabilities that require basic local reasoning, they face challenges with more complex vulnerabilities found in real-world programs. In comparison to static analysis tools like CodeQL, LLMs fall short in terms of precision and accuracy. Additionally, they struggle to differentiate between vulnerable and patched code. Larger LLMs, such as GPT-4, are also vulnerable to adversarial attacks, where their performance mildly drops when malicious code is introduced. Furthermore, LLMs can be susceptible to data poisoning, where attackers inject harmful data into the training process, as well as backdoor attacks that manipulate the model during training to embed hidden vulnerabilities.

Despite these challenges, LLMs offer distinct advantages. They can explain their predictions in natural language, making it easier for users to understand the results. LLMs also excel in few-shot learning, meaning they can generalize from very few examples, which can be highly beneficial when limited training data is available. Additionally, using prompts specifically tailored to detect certain Common Weakness Enumerations (CWEs) can boost LLM performance. Incorporating advanced prompting strategies, such as step-by-step analysis, can further enhance their capabilities, enabling more accurate detection of software vulnerabilities.

IV. BERT for Code Processing

BERT can be effectively applied for embedding datasets by transforming raw text into high-dimensional vector representations. By leveraging its bidirectional architecture, BERT

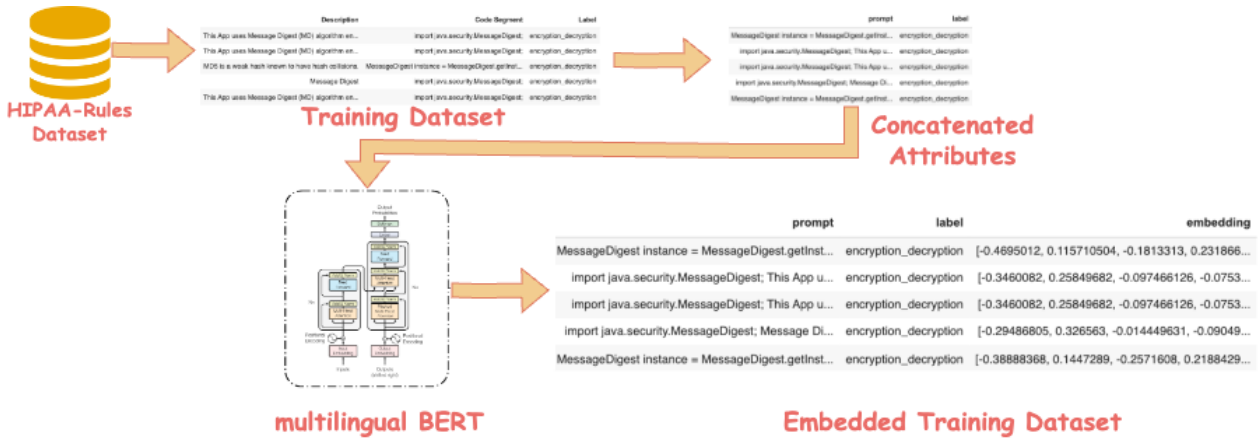

Fig. 4. Steps to embed prompt texts into vectors using a multilingual BERT model.

comprehends the context of each word in relation to the entire sentence, resulting in embeddings that reflect the nuances of language. This ability allows BERT to generate embeddings that retain important information, making them suitable for various natural language processing tasks. Fine-tuning BERT on specific datasets further enhances the quality of these embeddings, enabling models to better understand domain-specific terminology and relationships within the data.

When applied to embedding datasets, BERT utilizes its masked language modeling and next sentence prediction tasks during pretraining to create context-aware representations. These embeddings can be used in downstream applications such as text classification, code analysis, vulnerabilities detection and classification. By representing each piece of text as a dense vector, BERT facilitates the identification of semantic similarities and relationships between different data points. This capability allows for improved performance in various machine learning tasks, particularly in applications where understanding context and meaning is crucial.

V. Dataset

This research utilizes a unique dataset developed by the HIPAA lab at UBITRIX INTERNATIONAL, INC., as part of the STTR-II project. The project focuses on creating HIPAA compliance rule patterns for various platforms, including Android, iOS, and web applications. We have collected code samples of Android applications from our HIPAA engine codebase to create our own dataset, STTR-HIPAA, by our lab members. The dataset was specifically generated by testing Android applications and identifying matching code segments related to HIPAA guidelines. It contains four key attributes: ID, which uniquely identifies each record; Description, providing a brief explanation of the detected vulnerability; Code_Segment, which highlights the relevant portion of the Android code; and Label, indicating which of the HIPAA rules is being violated. The Label has 10 rules which are: user_inactivity, phi_encryption, encryption_decryption, data_integrity, authorization, auth_destruction, user_authentication, unique_id, transmition_secuirity, and audit. Fig. 3. shows the frequencies of observations for all HIPAA rules. The dataset comprises 252,384 observations, covering ten distinct HIPAA rules that govern the protection and security of sensitive healthcare data.

Also, we have collected codebase dataset from kaggle [33]. This dataset contains Java code snippets labeled by safety status—either ”safe” or ”vulnerable.” It supports vulnerability detection tasks by providing real-world code examples, helping train models to classify and understand secure versus insecure programming patterns in software projects. However, we have used only 400 instances for training ML models that is one reason to get highest number of True positive and True negative.

VI. Methods

Our methodology focuses on analyzing and classifying code patterns associated with various HIPAA rules present in source code. This process begins by collecting samples of code patterns from our codebase, specifically targeting segments that may violate HIPAA compliance. We label these code samples accurately, depending on their specific rule matches.

To ensure the quality of the created dataset, we employ machine learning techniques for further analysis. Specifically, we use an ”evaluating and correcting” approach to adjust the labels in the datasets based on the classification outcomes we obtain from our models. This step is crucial for maintaining the integrity of our dataset.

Once we assess whether an application complies with HIPAA, our HIPAA engine stores the relevant set of codes, indicating which rules have been matched. Initially, we create and compile the STTR dataset, which we then split into training and testing sets. The dataset is used with multilingual BERT to embed attributes (Fig. 4). This embedded dataset serves as the foundation for training four distinct types of machine learning model, each of which has been focused with optimized parameters (Fig. 5). After training, we tested all models to evaluate how it works to classify HIPAA rule code patterns effectively. This classification process is essential for software developers, enabling them to identify and rectify

specific areas within their applications that require compliance adjustments.

VII. Results and Discussions

A. Accuracy Metrics

Accuracy is quantified using the following formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **TP (True Positives)**: The number of positive cases correctly identified by the model.
- **TN (True Negatives)**: The count of negative cases accurately classified.
- **FP (False Positives)**: Negative instances mistakenly classified as positive.
- **FN (False Negatives)**: Positive instances that the model failed to identify as positive.

Precision evaluates the correctness of the model when it predicts positive cases:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall, on the other hand, measures the proportion of actual positive cases that the model successfully identifies:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The F1-score, which ranges from 0 to 1, provides a single metric to gauge model performance by balancing precision and recall. A score of 0 reflects poor performance, while a score of 1 indicates exceptional performance:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

These statistical measures are vital for evaluating the efficacy of the prompt injection detection system in recognizing malicious prompt attacks.

B. Results

In our research, we implemented DistilBERT to detect malicious prompts and multilingual BERT for embedding attributes. Our findings indicate that the multilingual BERT model generally outperforms other classifier when it comes to classifying HIPAA rules patterns as the BERT-base-multilingual-uncased model is specifically designed to handle multiple languages effectively. This model has been trained on a diverse range of texts across various languages, enabling it to comprehend and generate text embeddings suitable for different linguistic contexts. Consequently, this design leads to enhanced performance and improved accuracy when processing multilingual data, making it particularly adept at understanding nuanced text variations. In summary, our analysis highlights the importance of selecting the right machine learning model for specific tasks, particularly in the context of multilingual code patterns processing for classification.

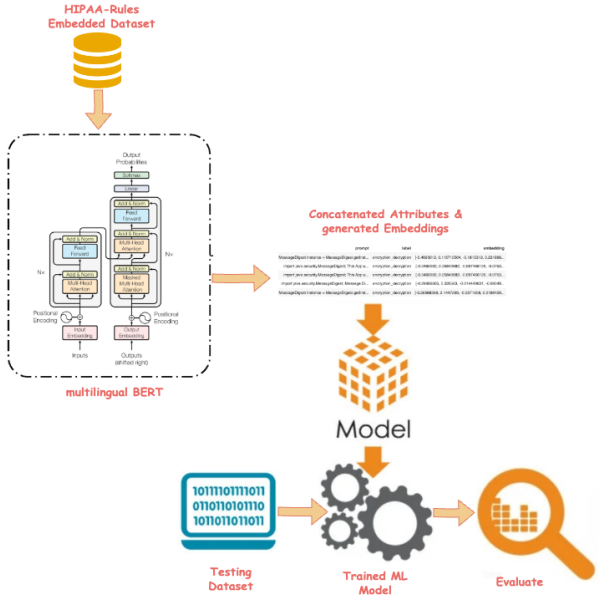

Fig. 5. The proposed architecture to embed dataset into vectors using pre-trained BERT for training various ML models for classification.

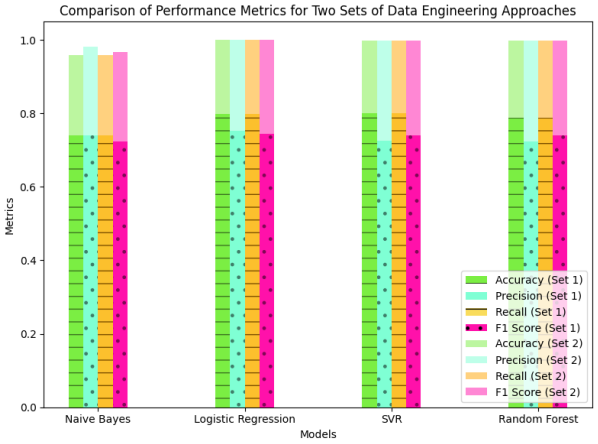

Fig. 6. Bar diagram of Accuracy Metrics based on Data Engineering.

The performance metrics presented in TABLE I indicate that the machine learning models applied solely to the 'Code Segment' attribute of the direct dataset yielded moderate results. In other words, when we did not concatenate the 'Description' attribute with the 'Code Segment' and performed no data engineering, the model performance remained limited. The highest recorded accuracy was 0.8011 for Support Vector Regression (SVR), while other models, such as Logistic Regression and Naive Bayes, displayed lower accuracy and precision scores. The overall F1 scores, which ranged from 0.7228 to 0.7440 suggest that the models struggled to accurately identify true positives.

In contrast, TABLE II highlights significant enhancements in model performance after applying data engineering techniques to the dataset. The metrics show impressive accuracy, with Logistic Regression achieving 0.9995 and SVR also

demonstrating high precision and recall. The corresponding F1 scores reflect this improvement, soaring to 0.9995. Additionally, a bar diagram visually compares these two sets of performance metrics, clearly illustrating the dramatic improvements achieved through data engineering (Fig. 6). The enhanced metrics underscore the vital role of preprocessing in optimizing machine learning models and their ability to deliver reliable predictions, emphasizing the importance of quality data in model training.

TABLE I
PERFORMANCE METRICS OF VARIOUS ML MODELS (ONLY CODE SEGMENT ATTRIBUTE)

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.7401	0.7392	0.7401	0.7228
Logistic Regression	0.7986	0.7523	0.7986	0.7440
SVR	0.8011	0.7265	0.8011	0.7400
Random Forest	0.7876	0.7232	0.7876	0.7395

TABLE II
PERFORMANCE METRICS OF VARIOUS ML MODELS (CONCATANATION OF TWO ATTRIBUTES)

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.9592	0.9815	0.9592	0.9667
Logistic Regression	0.9995	0.9995	0.9995	0.9995
SVR	0.9978	0.9978	0.9978	0.9978
Random Forest	0.9972	0.9972	0.9972	0.9972

TABLE III
COMPARISON THE PROPOSED BERT AND EXISTING WORKS

ML	Dataset	BERT	Acc.
LSTM	Fake or Real News	X	80.54
HDSF (Karimi [28])	Fake or Real News	X	82.19
TCNN	Weibo	X	88.08
TCNN-URG (Qian [29])	Weibo	X	89.84
Deep Learning (Tsinganos [30])	CSE-Persistence	CSE-Pers-BERT	86.27
Deep Learning	**Prompts-injection**	**DistilBERT**	**63.76**
Logistic R. [20]	**Prompts-injection**	**Multilingual**	**96.55**
Logistic R. [33]	**CVEFixes.csv**	**CodeBERT**	**99.91**
Proposed	**STTR-HIPAA**	**Multilingual**	**99.95**

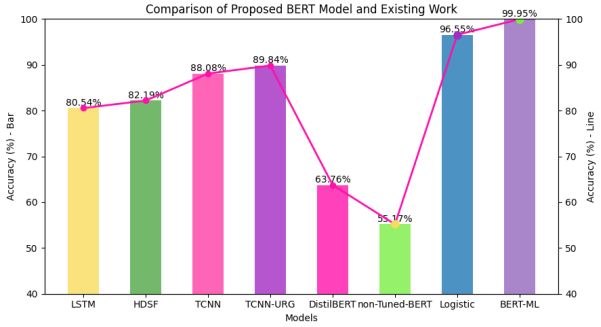

Fig. 7. Comparison Proposed BERT Model and Existing Works.

Table III compares the accuracy between our proposed model and existing works. The existing models include LSTM and HDSF applied to the "Fake or Real News" dataset, which achieved accuracies of 80.54% and 82.19%, respectively. The TCNN and TCNN-URG models, evaluated on the Weibo dataset, demonstrated a slightly better performance, reaching accuracies of 88.08% and 89.84% (Fig. 7).

In contrast, the Logistic Regression model using CodeBERT on the CVEFixes.csv dataset achieved 99.91% accuracy, showcasing strong performance. In comparison, the proposed model, leveraging Multilingual BERT on the STTR-HIPAA dataset, slightly outperformed it with 99.95% accuracy, suggesting enhanced generalization across diverse textual and code-based inputs. This remarkable enhancement underscores the efficacy of our proposed model in achieving superior performance compared to previous works, including those using DistilBERT and Logistic Regression, which attained lower accuracy rates of 63.76% and 96.55%, respectively for prompt injection attacks detection. Our results not only emphasize the potential of multilingual BERT for handling diverse datasets but also showcase its superiority in accurately detecting malicious prompts. Overall, the proposed model demonstrates a substantial advancement over existing methods, solidifying its position as a leading approach in the field. The weakest part of our dataset was using Description attribute in which code related information was available that helps to models to classify the HIPAA rules more accurately.

VIII. CONCLUSION

We found the challenges of HIPAA rules classification and identified the limitations of traditional Word2Vec embeddings. To address this, we processed attributes through data engineering and applied multilingual BERT to enhance the classification of the rules patterns using various ML algorithms. After applying data engineering, the Logistic Regression reached an impressive accuracy of 99.95%. Other models, including SVR and Random Forest, also demonstrated high performance, highlighting the effectiveness of multilingual BERT in promoting secure and compliant application development compared to exiting detection approaches. This work underscores the potential for improved security and compliance in the development of healthcare-related applications. We would like to fine the pre-trained LLMs with specific dataset for detection whether the applications are HIPAA compliant or not.

ACKNOWLEDGMENT

This research is funded by the National Science Foundation (NSF) and the National Institutes of Health (NIH) (Award # 5R42LM014356-03). Also, this work is supported by the National Science Foundation under NSF Awards #1946442 and #2433800. Any opinions, findings, recommendations, expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Lookout, "Mobile Threat Landscape Report: 2023 in Review," 2023. Available: https://www.lookout.com/threat-intelligence/report/mobile-landscape-threat-report, last visited 10/3/2024.

[2] OWASP, "Mobile Top 10 2024: Final Release Updates," 2024. Available: https://owasp.org/www-project-mobile-top-10/, last visited 10/3/2024.

[3] V. Kouliaridis, G. Karopoulos, and G. Kambourakis, "Assessing the security and privacy of android official id wallet apps," Information, vol. 14, no. 8, 2023.

[4] K. W. Church, "Word2vec," Natural Language Engineering, vol. 23, no. 1, pp. 155–162, 2017.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019.

[6] I. Solaiman et al., "Release strategies and the social impacts of language models," 2019.

[7] OpenAI, "OpenAI," 2024. Available: https://openai.com/, last visited 10/3/2024.

[8] T. B. Brown et al., "Language models are few-shot learners," 2020.

[9] Y. Wan, W. Zhao, H. Zhang, Y. Sui, G. Xu, and H. Jin, "What do they capture? A structural analysis of pre-trained language models for source code," in Proceedings of the 44th International Conference on Software Engineering, ICSE '22, New York, NY, USA, pp. 2377–2388, 2022, Association for Computing Machinery.

[10] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, "Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation," 2023.

[11] Hugging Face, "Hugging Face," Online. Available: https://huggingface.co/, accessed on April 16, 2024.

[12] M. S. Akter et al., "HIPAA Technical Compliance Evaluation of Laravel-based mHealth Apps," in 2024 IEEE International Conference on Digital Health (ICDH), pp. 58-67, 2024.

[13] A. B. K. Riad et al., "Enhancing HIPAA Compliance in AI-driven mHealth Devices Security and Privacy," in 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 2430-2435, 2024.

[14] M. A. Rahman, "Detection of Distributed Denial of Service Attacks Based on Machine Learning Algorithms," International Journal of Smart Home, vol. 14, no. 2, pp. 15-24, 2020.

[15] S. Akter, M. A. Rahman, S. Hossain, and M. Rahman, "Early Prediction of Cryptocurrency Price Decline: A Deep Learning Approach," 26th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, Dec. 2023.

[16] M. A. Rahman and S. Hossain, "Clustering Enabled Robust Intrusion Detection System for Big Data using Hadoop-PySpark," 2023 IEEE 20th International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET), Boca Raton, FL, USA, Dec. 2023.

[17] M. A. Rahman and S. Hossain, "Towards Developing Generative Adversarial Networks based Robust Intrusion Detection Systems for Imbalanced Dataset using Hadoop-PySpark," 2024 International Conference on Innovations in Computing Research (ICR'24), Athens, Greece, Aug. 2024.

[18] M. A. Rahman et al., "A Quantum Generative Adversarial Network-based Intrusion Detection System," in 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1810-1815, 2023.

[19] M. A. Rahman et al., "Fine-tuned Variational Quantum Classifiers for Cyber Attacks Detection based on Parameterized Quantum Circuits and Optimizers," in 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), 2024.

[20] M. A. Rahman, H. Shahriar, F. Wu and A. Cuzzocrea, "Applying Pre-trained Multilingual BERT in Embeddings for Improved Malicious Prompt Injection Attacks Detection," 2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings), Mt Pleasant, MI, USA, 2024, pp. 1-7, doi: 10.1109/AIBThings63359.2024.10863664.

[21] M. A. Barek et al., "Mitigating Insecure Outputs in Large Language Models (LLMs): A Practical Educational Module," in 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 2424-2429, 2024.

[22] M. S. Akter et al., "Authentic Learning Approach for Data Poisoning Vulnerability in LLMs," in 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1504-1505, 2024.

[23] A. Cuzzocrea and S. Chakravarthy, "Event-based lossy compression for effective and efficient OLAP over data streams," Data & Knowledge Engineering, vol. 69, no. 7, pp. 678-708, 2010.

[24] A. Cuzzocrea et al., "A distributed system for answering range queries on sensor network data," in Third IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 369-373, Mar. 2005.

[25] A. Cuzzocrea, V. Russo, and D. Sacca, "A robust sampling-based framework for privacy preserving OLAP," in Data Warehousing and Knowledge Discovery: 10th International Conference, DaWaK 2008, Turin, Italy, Sep. 2008, pp. 97-114, Springer Berlin Heidelberg.

[26] A. Cuzzocrea, D. Saccà, and P. Serafino, "A hierarchy-driven compression technique for advanced OLAP visualization of multidimensional data cubes," in Data Warehousing and Knowledge Discovery: 8th International Conference, DaWaK 2006, Krakow, Poland, Sep. 2006, pp. 106-119, Springer Berlin Heidelberg.

[27] M. Cannataro et al., "Modeling Adaptive Hypermedia with an Object-Oriented Approach and XML," WebDyn@ WWW, pp. 35-44, 2002.

[28] H. Karimi and J. Tang, "Learning hierarchical discourse-level structure for fake news detection," arXiv preprint arXiv:1903.07389, 2019.

[29] R. Oshikawa, J. Qian, and W. Y. Wang, "A survey on natural language processing for fake news detection," arXiv preprint arXiv:1811.00770, 2018.

[30] N. Tsinganos, P. Fouliras, and I. Mavridis, "Applying BERT for early-stage recognition of persistence in chat-based social engineering attacks," Applied Sciences, vol. 12, no. 23, 2022.

[31] Rahman, M. A., Francia, G. A., Shahriar, H. (2025). Leveraging GANs for Synthetic Data Generation to Improve Intrusion Detection Systems. Journal of Future Artificial Intelligence and Technologies, 1(4), 429-439.

[32] Rahman, M. A., Francia, G. A., Shahriar, H. (2025, May). Large Language Model can Reduce the Necessity of using Large Data Samples for Training Models. In 2025 IEEE Conference on Artificial Intelligence (CAI). IEEE.

[33] Dataset (CVEFixes.csv) from Kaggle. https://www.kaggle.com/code/gustavoaca1997/oss-vulnerabilities-detection-notebook/input