Multimodal Strategy To Defend Mobile Devices Against Vishing Attacks

Manas Mishra, Saurabh Jain, Suraj Patel M, Sharmila Mani, Abhijeet Boragule, Nikhil Sahni and Renju Chirakarotu Nair
Samsung R&D Institute India - Bangalore
Email: {mishra.ms, jain.saurabh, suraj.m4, sharmila.m, abhijeet.y, nikhil.sahni and renju.cn}@samsung.com

Abstract

In the landscape of Vishing calls, it's crucial to determine if users are unknowingly falling prey to fraudsters' tactics, potentially leading to fraud. This paper introduces the Multimodal Vishing Threat Detection (MmVTD) framework, designed to counter Vishing threats and issue alerts on potential fraudulent activities leveraging three key data modalities: call transcripts, sequences of mobile screenshots, and extracted Optical Character Recognition(OCR) text. The novelty of our approach lies in utilization of transformer encoders to build the learnable context for each modality, with a decoder generating warning content while an MLP head classifies the current call as Vishing or Safe at any given instance. This innovative architecture enhances the model's ability to analyze the diverse data sources and generate effective warning in real-time. Furthermore, the MmVTD framework offers robust protection through multimodal analysis, providing real-time alerts when user action suggest susceptibility fraud. Finally, the MmVTD model achieved an impressive 94.44% accuracy in identifying Vishing and Safe calls on a dedicated dataset. Additionally, it attained a BLEU score of 0.583 in generating cautionary messages, effectively deterring users from potential harmful actions.

CCS Concepts

• **Security and privacy** → **Phishing**; **Intrusion detection systems**; • **Computing methodologies** → *Information extraction*; *Neural networks*; Supervised learning by classification.

Keywords

Vishing, Mobile Security, MultiModal Deep learning, Behavior Analysis, Security and Privacy, SmartPhone

ACM Reference Format:
Manas Mishra, Saurabh Jain, Suraj Patel M, Sharmila Mani, Abhijeet Boragule, Nikhil Sahni and Renju Chirakarotu Nair, Samsung R&D Institute India - Bangalore, Email: {mishra.ms, jain.saurabh, suraj.m4, sharmila.m, abhijeet.y, nikhil.sahni and renju.cn}@samsung.com . 2024. Multimodal Strategy To Defend Mobile Devices Against Vishing Attacks. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3636534.3690683

1 Introduction

The term Phishing has become synonymous with attempts to deceive individuals into revealing sensitive information, typically over web interfaces [1–4]. Voice Phishing attacks, known as Vishing attacks, are a category of Phishing attack where the attacker attempts to capture sensitive information through phone call [1, 3–5]. In an era marked by relentless digital advancement, Vishing takes Phishing deception to a more intimate level by using voice as a vector to impersonate trusted entities such as banks, government agencies, or even friends and family, and persuading victims to disclose personal or financial information [4, 6]. Therefore, it is imperative to not only acknowledge this threat but also device comprehensive and effective strategies to counter it.

To address Vishing attack challenges, we utilize the power of mulitiple data modalities which combines various forms of communication and data analysis. It proves to be superior in the realm of detecting and preventing Vishing attacks compared to individual modalities such as call audio recordings, call transcripts, and real time screenshots. The synergy of diverse modalities enables a more comprehensive contextual understanding of user behaviour and network patterns. For instance, combining voice and text analysis can provide a deeper insight into the context of communication, allowing for more accurate and deterministic threat identification. Whereas, only call transcript, audio recordings, or screen images may not provide the deterministic predictions. During the voice call, detecting abnormalities is challenging because of the human cognitive nature to explore various applications in smartphone dynamically. Therefore, the proposed solution combines all the three modalities together to caution the user in times of attack.

Utilising multimodality[7–10] data to solve a problem is experimented on different domains like voice recognition

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11.
https://doi.org/10.1145/3636534.3690683

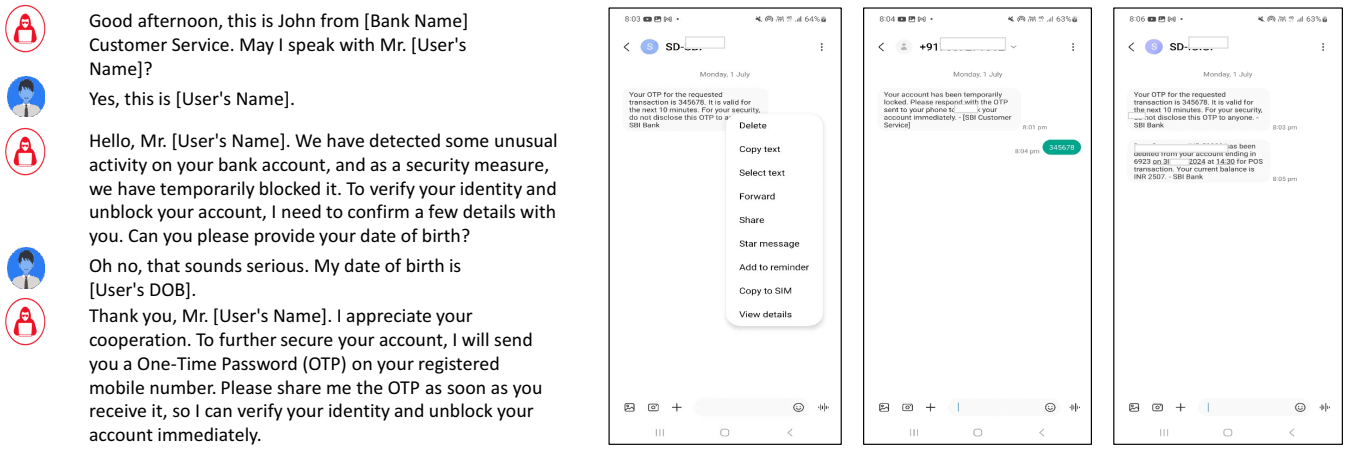

Figure 1: Illustration of a real world use case where attacker pretends to be a banker over voice call, requests user for one-time password required to unlock his account, shows: a) User receiving an one-time password from his bank for completing a pending transaction. b) User action performing copy and paste of OTP at the caller's chat. c) Receives a confirmation of a successful transaction from his bank.

[11, 12] , emotion recognition [13–15], medicine [16–18], education [19–21], including Vishing [22]. Unlike Kedem et al. [22], we are first to introduce a method to combat Vishings attacks using modified Transformer architecture [23] and multimodalities including call transcripts, smartphone screenshots, and OCR text from screenshots. More specifically, we introduce four independent encoders to learn the context from the different modalities. Furthermore, a classifier head to classify vishing attacks. Finally, a decoder to generate warning message to user about ongoing smartphone activity during phone call. This end-to-end pipeline provides a holistic approach for identifying threat during call and alert the user to take further steps.

This paper aims to provide a multimodal solution to the pressing problem of Vishing attacks (illustration of an example usecase in Figure 1) and our contributions are summarised as follows.

- An end-to-end system which utilises modified transformer architecture for multimodal data to identify Vishing attacks.
- We introduce separate encoders for each modality to learn the contextual information followed by classification layer and decoder to generate caution message.
- Concatenation of multimodal data is compensated by the Fully Connected (FC) layer to address the dynamic nature of multimodal data.
- The evaluation on collected dataset shows that our proposed method have demonstrated significant performance.

This section briefly introduced the problem of Vishing attacks and our motivation. In the following sections, we elaborate the related works in Section 2, proposed methodology in Section 3, while Section 4 explains details of our experimental results on the proposed architecture. Finally, we conclude our research in Section 5.

2 Related Work

2.1 Vishing Attacks

To detect the Vishing Attacks, there have been tremendous efforts in research community to identify and mitigate the attack [24–26]. In 2-Path Hybrid Model [24], the approach identifies escalating malicious spam calls by utilising call transcripts using Universal Sentence Encoder. Brabin et al. [25] proposes a solution by using a Central Banking Server (CBS) as an authentication server and a nationwide phone number common to all, aiding the users to identify between a fake and a genuine call from the bank. Choi et al.[26] employs a crime script analysis to delve into the intricate stages of voice Phishing by examining a total of 182 case files and conducting interviews to reveal its unique characteristics. This study outlines the preparation, recruitment, and execution phases of Vishing. This work has contributed foundational insights for future research and policy implementation.
During pandemic, RIVPAM [27] addressed the increased prevalence of Vishing attacks through real-time prediction and awareness-raising about prospective Vishing risks. The RIVPAM seeks to provide a proactive defense towards detection and prevention of such attacks. Jones et al. [28] suggests a three-dimensional examination that includes neurophysiological measurements including electroencephalograms (EEGs), eye gaze patterns and cognitive metrics in addition

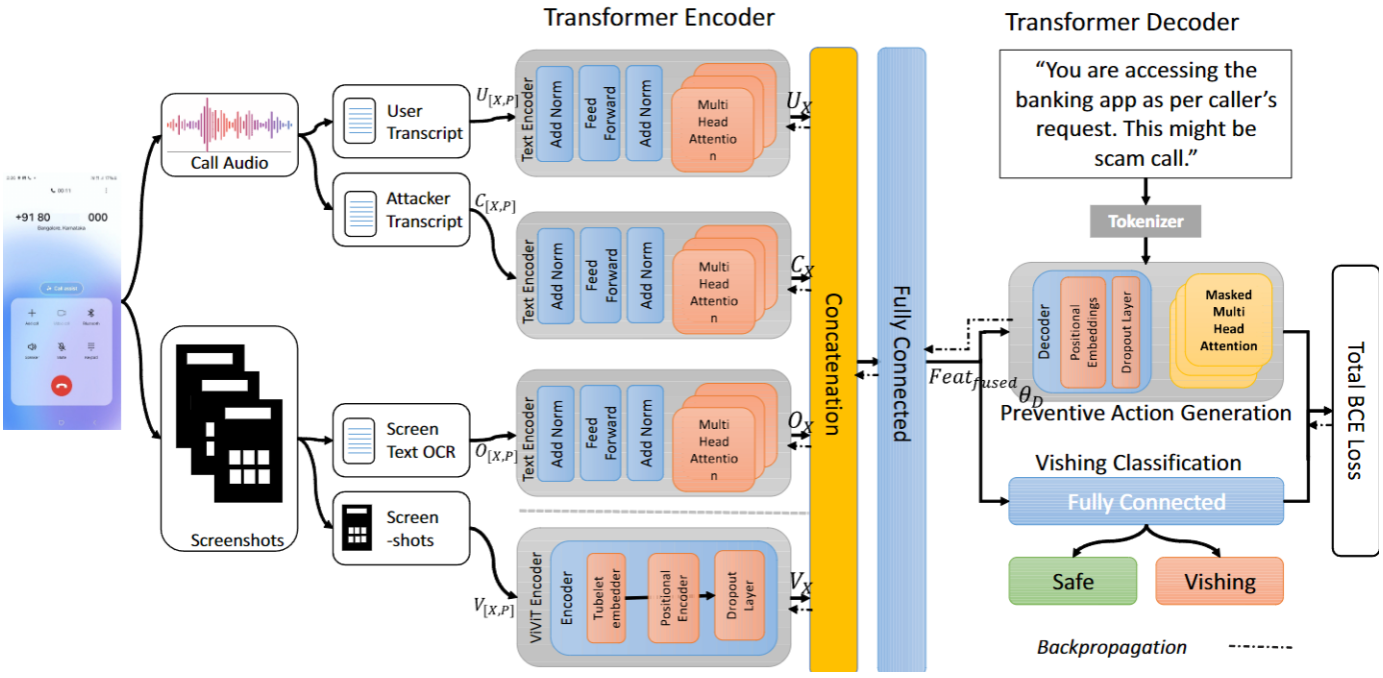

Figure 2: MmVTD Architecture : This is designed to process three distinct input signals: call transcripts, screen text obtained through optical character recognition and screenshots. These inputs are sent into their specific encoders for feature extraction. Subsequently, a block of concatenation and fully connected layer is used to fuse all the extracted features. Subsequently, $Feat_{fused}$ is directed to the Vishing Classification block for classification and Preventive Action Generation block for text alert generation. The classification and generation losses are combined to create an overall loss that is eventually utilized to train the system parameters

to conventional task performance data. This study sheds light on user behavior during Phishing detection activities and highlights difficulties in analyzing Phishing indicators since users may be cognitively focused on the job at hand but have difficulty identifying important Phishing cues. Lee et al. [29] suggest utilizing fundamental machine learning models to develop a real-time detection method in Korean language. Vishing audio samples are converted into text as part of the study and natural language processing techniques have been applied to detect the importance of quick detection over careful model development. This motivated us to extract and preprocess call transcripts for our model solution. In the Vishing attacks challenges, different from previous works, MmVTD introduces a multimodal transformer architecture to combat Vishing attacks.

2.2 Multimodal Approaches

In the Transformer era, various methods and architectures for multimodal fusion have been extensively explored, leading to new approaches [30]. The Audio-Visual Speech Recognition(AVSR)[31] explores a methodology where uni-modal deep networks are individually trained and their final hidden layers are fused to establish a joint feature space. This fusion model significantly reduces the Phone Error Rate (PER) compared to standard audio networks, indicating the substantial role of the visual channel in phone classification even with the high signal-to-noise ratio. Yoon et al. [32], addresses the challenge by proposing a deep dual recurrent encoder model that also utilizes both text data and audio signals. This model, employing dual Recurrent Neural Networks (RNNs), outperforms benchmark methods on emotional dialogue, achieving accuracies between 68.8% and 71.8% across detecting anger, happiness, sadness, and neutrality.

The methodologies presented in these suggest a limited approach where voice transcripts, authentication servers, human strategies and call duration serve as the only basis for detecting Vishing calls. Our proposed methodology uses voice transcripts with ongoing user patterns when accounting for and forecasting Vishing assaults by combining the modalities from the singular model.

3 Proposed Methodology

Our proposed method aims to enhance the detection accuracy of Vishing attacks in smartphones by leveraging sophisticated multimodel data and multi-input architecture that integrates both textual and visual information extracted during phone call. At its core, the architecture comprises to process and extract meaningful features from various sources of data. Firstly, transformer encoders employed to process the caller's transcript text and user's transcript text, converting the spoken dialogue into structured textual representations. Secondly, an encoder is utilized to extract text from screenshots captured during the call using OCR techniques, enabling the analysis of any visual information present in the call interface. Thirdly, a ViViT encoder is employed to extract relevant features from the screenshots, capturing visual patterns and contextual information that may indicate fraudulent activity is ongoing as per attacker instructions as depicted in Figure 2.

These components collectively generate encoded representations and feature vectors, which are then concatenated into a single comprehensive representation. This concatenated representation is subsequently passed through a fully connected layer, which servers to fuse the extracted information and produce a shared representation. This representation is utilized for two primary purposes: generating warning messages to alert users of potential Vishing attacks, and performing classification to categorize the nature and severity of detected threats. By integrating information from multiple sources and leveraging advanced neural network techniques, our method aims to provide robust and real-time detection capabilities to protect smartphone users from increasingly sophisticated Vishing attacks.

3.1 Preliminaries

To begin, we outline the initial preliminaries necessary for our approach. Let X_{caller} denote the caller transcript text after tokenization. Its dimensionality can be represented $X_{caller} \in \mathbb{R}^{T_{caller} \times D_{token}}$, where T_{caller} represents the number of tokens in the caller transcript, and D_{token} represents the dimensionality of each token embedding. Similarly, let X_{user} denote the smartphone user's transcript text after tokenization. Its dimensionality can be represented $X_{user} \in \mathbb{R}^{T_{user} \times D_{token}}$, where T_{user} represents the number of tokens in the caller transcript, and D_{token} represents the dimensionality of each token embedding. Moreover, let X_{OCR} denote the input screen texts after tokenization. Its dimensionality can be represented $X_{OCR} \in \mathbb{R}^{T_{OCR} \times D_{token}}$, where T_{OCR} represents the number of tokens in the OCR text, and D_{token} represents the dimensionality of each token embedding similar OCR text embedding representation. Finally, let I_{scr} denote the input image from screenshots after preprocessing. Its dimensionality can be represented as $I \in \mathbb{R}^{H \times W \times C}$, where H and W represent height and width of the image respectively, and C represents the number of channels.

3.2 User, Caller, and OCR text encoder

Each encoder follows a standardized procedure: taking tokenized text inputs and generating fixed-dimensional numerical representations, ensuring consistency across the encoding process. These encoded representations serve as the foundation for subsequent analysis or downstream tasks, providing a comprehensive understanding of the conversation and image content. Through this systematic encoding approach, we enable efficient data processing and robust feature extraction, facilitating advanced analysis and decisionmaking processes.

The user, caller and OCR encoder output has been computed as follows,

$$\begin{aligned} UX &= EncoderUser\left(X_{user} + P_{user}\right) \\ CX &= EncoderCaller\left(X_{caller} + P_{caller}\right) \\ OX &= EncoderOCR\left(X_{OCR} + P_{OCR}\right) \end{aligned} \tag{1}$$

where X_{user} ,X_{caller}, and X_{OCR} are input tokenized embeddings and P_{user}, P_{caller} and, P_{OCR} are the positional encoding of text encoding.The UX, CX, and OX are leaned local and global contextual representation. These hidden states represent the encoded information of the corresponding input tokens in the sequence [23]. The encoders are independent and do not share any parameters among them.

3.3 Visual Encoder

We leverage the well-established ViViT encoder [33] as a fundamental framework for capturing the intricate visual flow of user interactions with the device screen. However, the exploration incorporates the use of tubelet embedding , renowned for its claim to encapsulate spatiotemporal understanding across diverse sequential image inputs. The resulting feature arrays of sequence screen images are revealed to the deployment of two key components for embedding generation: the tubelet embedding layer, which captures temporal features and the positional embedding layer, which embeds an understanding of the relative positions of various patches. Importantly, the embedding process is structured to create a combined 3-D array forming a thorough picture of the spatial and temporal dynamics of the changing user interaction sequence. It is noteworthy that, beyond this modification, the architecture adheres to the fundamental ViViT encoder design, underscoring the adaptability and efficiency of the proposed framework in enhancing encoder performance for analysis of sequential screen images. The visual encoding is computed as below,

$$VX = EncoderVisual\left(I^{H \times W \times C}\right) \tag{2}$$

where, I is the input image screenshot with dimension as $H \times W \times C$ and VX are the learned hidden states of encoder which represents contextual representation.

The decision to use ViViT was based on its ability to handle both temporal and spatial information, which is essential given that mobile screenshots form a sequence of images capturing interactions over time. Traditional Convolutional Neural Networks (CNNs) excel at spatial feature extraction but lack temporal modeling capabilities, while Recurrent Neural Networks (RNNs) handle temporal sequences well but are less effective with spatial details. ViViT overcomes these limitations by leveraging the transformer architecture, which excels in both areas. The transformer architecture allows for parallel processing of input sequences, enhancing efficiency, which is critical for real-time vishing threat detection. Additionally, ViViT's attention mechanism improves the detection of subtle cues indicative of vishing attempts. ViViT also offers scalability and flexibility, handling longer sequences more effectively and integrating seamlessly with other modalities in our framework, such as call transcripts and OCR text.

3.4 Multimodal Fusion

In our framework, multimodal fusion plays a pivotal role in integrating information from diverse sources, enabling a comprehensive understanding of the conversation content. By concatenating the outputs of textual and visual encoders along a specified axis, we create a unified representation that encapsulates both semantic and contextual information. This fusion mechanism allows the model to leverage the complementary strengths of different modalities, enhancing its ability to capture nuanced aspects of the conversation. Whether it's textual semantics captured by user and caller encoders or contextual cues extracted from screenshots by the visual encoder, the fusion process ensures that all relevant information is seamlessly integrated, paving the way for more robust and informative representations.

Synthesis of multimodal fusion and fully connected Layer. Together, multimodal fusion and the fully connected layer form a powerful framework for processing and integrating information from diverse modalities. The fusion mechanism ensures that all relevant information is effectively combined into a unified representation, while the fully connected layer facilitates the learning of complex relationships within this representation. This synergy between fusion and learning enables the model to extract rich and informative features from multimodal inputs, leading to improved performance across various downstream tasks. Ultimately, the combination of multimodal fusion and the fully connected layer empowers our framework to achieve a comprehensive understanding of conversation content, even in the presence of dynamic inputs or missing modalities.
The multimodal fusion is computed as follows,

$$Feat_{fused} = f\left([UX, CX, OX, VX]\right) \tag{3}$$

where $Feat_{fused}$ are the fused multimodal features. The UX, CX,OX and VX are the encoded hidden-state features.

3.5 Caution generation with decoder

The output from the fully connected layer servers as main input for two crucial tasks within our frameworks: caution message generation about ongoing smartphone activity and Vishing attack classification. Firstly, for caution message generation, the fused representation from the fully connected layer is fed into a transformer decoder. This decoder leverages masked self-attention mechanism to learn the context against the caution message and multimodal fused features. The decoder generates cautionary messages tailored to conversation context and alert the smartphone users about potential risk. The transformer decoder generates the textual contents as follows,

$$T_{xt} = Decoder\left(Feat_{fused}, Tx_{t-1}\right) \tag{4}$$

where, Tx_t is the generated word and $Feat_{fused}$ is the fused features from the FC layer. Additionally, the output from the fully connected layer is utilized for Vishing attack classification, a task aimed at detecting and mitigating fraudulent or malicious content within the conversation. By leveraging the rich multimodal features encoded in the fused representation the classification model can identify suspicious patterns indicative of Vishing attacks, such as deceptive language, manipulative tactics. The classification is determined as follows,

$$Decision = cls\left(Feat_{fused}\right) \tag{5}$$

The cls layer contains an additional fully connected layer which outputs the binary classification about Vishing call at given instance.

3.6 Total Loss

In our framework, the total loss function is a critical metric that encapsulates the combined loss from the both decoder and classification tasks, providing a comprehensive measure of model performance. Firstly, the decoder loss $\mathcal{L}_{decoder}$ is calculated using standard sequence-to-sequence loss functions, such cross-entropy, to measure the discrepancy between the predicted caution messages and the ground truth. This loss component evaluates the accuracy of the generated caution messages, ensuring they closely match the intended content. Additionally, the classification loss $\mathcal{L}_{cls}$ is employed to assess the model's performance in distinguishing between legitimate conversation content and potential Vishing attacks. This loss, typically computed using binary cross-entropy which quantifies the model's ability to detect

Vishing attacks. By summing those losses, each weighted by a coefficient $\lambda_{decoder}$ and λ_{cls} respectively, the total loss $\mathcal{L}_{total}$ provides a unified metric for optimizing the model parameters and guiding the training process. Mathematically, the total loss can be expressed as:

$$\mathcal{L}_{total} = \lambda_{decoder} * \mathcal{L}_{decoder} + \lambda_{cls} * \mathcal{L}_{cls} \quad (6)$$

where $\lambda_{decoder}$ and λ_{cls} are both equal to 0.90 since they follow same units in cross entropy.

The decision to use a Total Loss metric combining classification and decoder losses is based on several key considerations that make it more appropriate for evaluating the performance of the MmVTD framework. By unifying classification and decoder losses into a single Total Loss metric, both tasks are optimized simultaneously, ensuring consistency and alignment of the model's objectives. This integrated approach ensures the MmVTD framework effectively classifies calls as Vishing or safe while also generating effective warning messages. The mutual reinforcement between classification and text generation tasks is facilitated by the Total Loss metric, promoting balanced optimization and preventing the model from prioritizing one task over the other. In practical scenarios, users benefit from accurate classification and high-quality warnings simultaneously, making the Total Loss metric relevant for real-world applications and directly impacting user experience by ensuring both aspects are addressed. Additionally, combining losses simplifies the training process by providing a single objective, leading to more efficient training and convergence. This holistic evaluation makes it easier to assess and compare different model versions or configurations, ultimately enhancing the overall performance of the MmVTD framework.

4 Experiments and Evaluation

4.1 Dataset Overview

This section details on steps involved in data collection and preprocessing done exclusively for MmVTD. Firstly, collection of Vishing call transcript data is a crucial aspect that aids in capturing context of the call. This data not only offers insights into the conversational dynamics but also proves instrumental in differentiating between personal and professional calls, thereby enhancing the contextual understanding that helps in accurate classification of the call as Safe or Vishing. Secondly, our approach integrates the analysis of sequence of screen images captured during Vishing calls. These images serve as visual cues, reflecting the intricate details of the user's interactions on the device screen throughout the call. The visual information is paramount for identifying anomalies or manipulative tactics employed by fraudsters. Finally, we augment our analysis by incorporating sequences of Optical Character Recognition (OCR) text derived from the same set of images. This addition enriches the visual understanding of user interactions extracted from the sequence of screen images by providing insights into the associated textual content. By employing this triad of data modalities, our methodology aspires to offer a holistic and nuanced approach to real-time Vishing attack detection.

Our initial focus involved the meticulous creation of a dataset that authentically mirrors real world Vishing scenarios. This dataset is curated to capture the nuances and complexities of genuine Vishing attempts as below.

- Safe - User receives a Safe call and interacts with the phone for same or different purpose
- Safe - User receives a Safe call and does not interact with ones phone
- Safe – User receives a Vishing call and does not interact with phone / interact for different purpose
- Vishing - User receives a Vishing call and interacts with one's phone as per caller or attackers intention

We have collected a total of 1771 instances, expertly generated and annotated by human assessors, where each instance has the corresponding data modalities that were previously described. If user does not interact with his/her phone during the call, then the corresponding screenshots and OCR are zero paded for training purpose. The flow diagram representing different modalities not present at different time intervals is depicted in Figure 3. A random sample of all 3 modality dataset has been augmented for scenarios to include with and without call, screen and OCR text and its combinations that increased our dataset by 7-fold i.e 12397.

4.2 Data Generation and Annotation

To facilitate the comprehensive construction and annotation of the acquired dataset, a purpose-built mobile application named *ULFApp (User Labeling Facilitator Application)* was employed. This application serves the dual purpose of capturing the diverse data modalities associated with Vishing calls and assisting in the precise annotation processes. A sample illustration of incoming call and user accepting the call and annotation after call is shown in Figure 4. ULFApp automatically captures required data modalities corresponding to the incoming voice call and enables annotators to categorize them as Vishing or non-Vishing upon call completion. Our data generation initiative involves the acquisition of audio data from a substantial number of Vishing and non-Vishing calls. To accomplish this, 40 proficient volunteers, each fluent in English language were enlisted. These volunteers are divided into two distinct teams - Team A and Team B. Team A acts as smartphone user or voice call receiver and engages in conversation with captures the activity using ULFApp while Team B acts as a caller with their own intention of safe or Vishing purpose to make the conversation more natural.

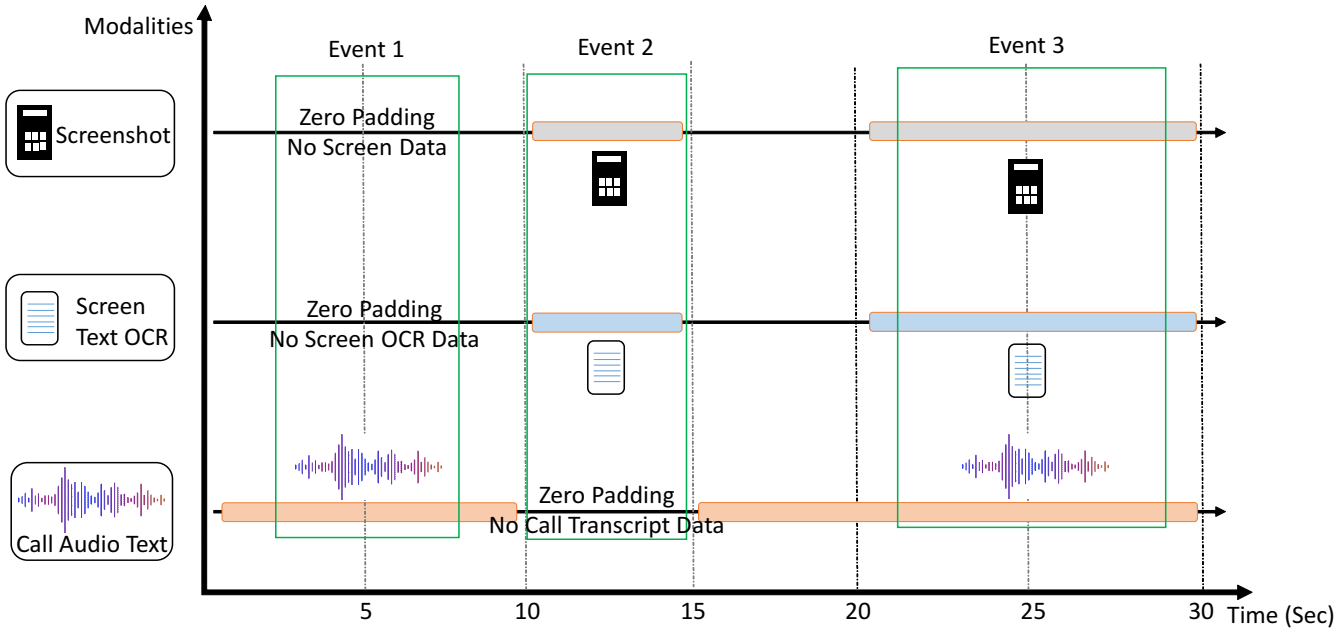

Figure 3: Flow Diagram:A sample depiction of MmVTD data collection flow in real time scenario. Event 1 only has the call transcripts; Event 2 is when user trying to access his device; Event 3 is when user is talking while operating the device.

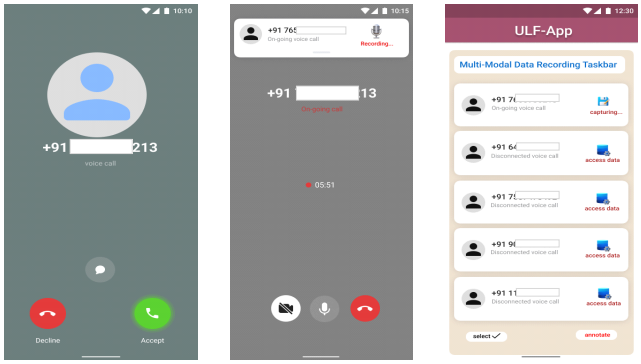

Figure 4: An illustration of the ULFApp data collection process shows: a) A volunteer receiving an unknown voice call; b) While in the background, ULFApp is gathering multimodal data related to the ongoing call in real time; and c) The volunteer using the ULFApp multimodal data taskbar to annotate data.

At the end of call, caller mentions intention of the call that is annotated by user in ULFApp.

ULFApp plays a vital role in recording on-screen user activity, capturing continuous screen frames triggered by user touch events. Importantly, frame capturing occurs only when the current frame exhibits a visual difference of at least 20% compared to its captured chronological predecessor screen frame. Specifically, we utilize *semantic texton forest (STF)* [34] to get similarity scores between screen images that are compared, and then we use *structural similarity index measure (SSIM)* [35] score of normalized similarity score and further translate them to a percentage scale using:

$$Percentage\ Score = \left(\frac{SSIM + 1}{2}\right) \times 100 \tag{7}$$

Each captured frame undergoes thorough processing within ULFApp, employing Google's ML-Kit Text Recognition [36] to extract content embedded in the screen.

Our next step involves the creation of call transcripts (call conversation text) for all the recorded call audios. To accomplish this, the ULFApp leverages Google Cloud's Speech-to-Text Automatic Speech Recognition (ASR) [37] service. These transcripts maintain the chronological order of the dialogues between the caller and the attendant, ensuring that the text reflects the sequence in which the conversations unfolded during the call. A team of 6 linguists played crucial role in constructing cautionary texts associated with each voice call, guided by the earlier annotated labels and leveraging various data modalities. In instances where the voice call did not exhibit Vishing characteristics or the user's actions were not influenced by the caller adversary, a standardized response of "Safe" was assigned by the linguists. To ensure

consistency and accuracy in cases with multiple annotated cautionary alerts linked to the same voice call, a supervisory team consisting of 2 members undertook the finalization process, contributing to the robustness and reliability of our cautionary phrase annotations. Finally, we store the associated call transcript, sequence of screen images, OCR text extracted from the screen images, and the annotated cautionary phrase in a dedicated folder with a unique name in *{phonenumber}_{timestamp}* format, where *phonenumber* is the caller's caller Id and *timestamp* is the moment it started recording.

4.3 Data Preprocessing and Analysis

4.3.1 Voice Call Transcript Data (VCTD). The voice call transcripts data underwent an deliberate clean-up process detailed in Figure 4 to enhance its quality and relevance. Initially, the irrelevant symbols, special characters, and punctuation marks are eliminated to ensure textual consistency. Subsequently, any residual non-text elements including timestamps, speaker labels and extraneous metadata, were removed to streamline the dataset for focused analysis. The challenges posed by abbreviation and named entities within the transcripts were handled systematically by anonymizing the elements to maintain privacy and coherence. Inspired by Munková et al [38], we implemented stop phrase removal further significantly reducing the average length of individual dialogues texts to 13.64 words. The next step involved padding the user's and caller's transcript separately to match the length of the longest text associated with their respective speaker types across the entire dataset. This comprehensive preprocessing ensures uniformity and cleanliness within the dataset but also helps in analysis of individual call transcripts.

4.3.2 Screen Activity Flow Data (SAFD). The Screen Activity Flow Dataset plays a pivotal role encompassing a rich array of data, including sequences of screen images accompanied by extracted OCR text. Recognizing the importance of optimizing computation efficiency and eliminating any remaining visual information, a judicial filtering process is implemented that is detailed in Figure 6. Alternate images were deliberately filtered out, starting the culling procedure from the chronological end of the sequences because the majority of the alternate images had overlapping pixel information. As a result of this processing, the maximum number of images per sequence was effectively reduced to 6. For sequences with fewer images or no screen information available, synthetic images with zero intensity pixels were introduced to pad the sequences to maintain uniformity. Subsequently, the computational overhead of SAFDs associated encoder were optimized by reducing the resolution of these images and corresponding heatmaps from 1080*2160 to 320*320. This multi-faceted approach not only streamlined the dataset but also enhanced the efficiency, preparing it for subsequent analyses and ensuring a balance between information richness and computation resource optimization.

4.3.3 OCR Text From SAFD. Despite the filtering process applied to screen images, we extract the OCR of images to get the textual uniqueness from visually overlapping images. To consolidate the extracted text, we aggregate to a comprehensive text file, setting the stage for further preprocessing. Similar to Voice Call Transcripts Dataset (VCTD), the text file is further passed through preprocessing steps. However, in this instance, an additional step to filter out redundant or repetitive textual phrases, enhancing the overall clarity of the extracted text.

4.4 Experiments

In our research, we executed a comprehensive series of experiments aimed at scrutinizing the efficacy of our proposed MultiModal Vishing Threat Detection (MmVTD) framework, rooted in the domain of multimodal deep learning. The primary objective of these experiments were twofold:

- To explore the impact of incorporating multiple data modalities on the generation of cautionary alert phrase.
- To ascertain whether the performance of MmVTD surpasses that of existing Vishing benchmarks.

The experimental process begins with a detailed discussion of our setup, encompassing the configuration of training parameters and model specifications. Subsequently, we present an examination of the performance of our models, employing a range of widely-adopted metrics to evaluate their effectiveness. This systematic approach ensures a robust evaluation of the MmVTD framework and contributes valuable insights into the landscape of Vishing threat detection in a multimodal context.

4.4.1 Implementation. The initialization of tokens plays a pivotal role in establishing a foundation for effective model training. Specifically, we employ pre-trained 400K-vocabulary 300-dimensional GLOVE embeddings [39] to initialize all tokens, excluding the unknown tokens. These embeddings are subsequently projected into a 128-dimensional vector space, providing a condensed yet information-rich representation. We adhere to specifications of the text encoders employed for representing the VCTD (Voice Call Transcript Dataset) and SAFD's (Screen Activity Flow Data) as in Section 3.2. Each encoder has a general hidden state of 256 and an output dimension of 128. For its configuration, we utilise a patch size of 16 and an embedding dimension 128 which aligns harmoniously with the output dimensions. We tested different setups with 8, 16, and 32 attention heads, and ultimately settled on 8 heads due to their performance and efficiency in terms of computing. This strategic choice aims to foster

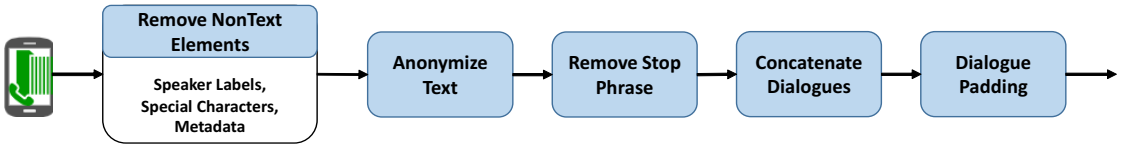

Figure 5: VCTD FlowDigram shows the call transcript data cleanup process in multiple stages to achieve higher quality and relevance

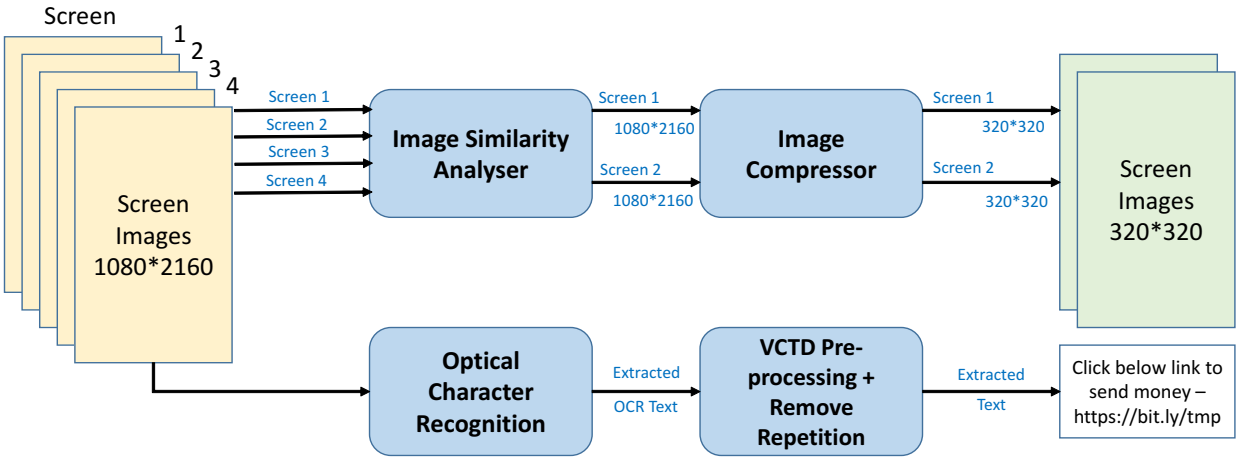

Figure 6: SAFD FlowDiagram explains the optimal filtering of screen images and their corresponding OCRs and further cleans up the OCR data to remove duplication

seamless integration of multimodal information, promoting effective cross-modal learning. The decoder detailed in Section 3.5 helps us to generate each word of the desired cautionary phrase sequentially. TensorFlow and Keras were used to build the complete framework, and 4 NVIDIA A600 GPU cores were used for further training along with a batch size of 32 and the Adam optimizer.

4.5 Ablation Studies

4.5.1 Data Modality Variants Comparison. In our pursuit to comprehensively examine the efficacy of integrating multimodal data representations for the generation of cautionary alerts for the user and vishing call classification, a systematic investigation was conducted by scrutinizing our trained model with variations of data modalities:

- *CallTranscriptOnly*: Includes caller and user transcripts
- *OCRTextOnly*: Focuses only on OCR Text extracted from Screen Images
- *SqScreenOnly*: Considers only Sequence of Screen Images
- *CallTranscript + SqScreen*: Helps to understand the combination of visual information from screen images and call transcript
- *SqScreen + OCRText*: Helps to understand the contextual information extracted from screen images and extracted OCR text
- *CallTranscript + OCRText*: the validity of call transcript with OCRText independent of screen images
- *CallTranscript + OCRText + SqScreen*: Collective information from CallTranscript, OCRText and Sequence of Screen Images

The same model, first trained across all three modalities, is employed for single modality evaluation with zero padding. The training dataset was deliberately adjusted to encompass data for the missing modality, ensuring a single solution is applicable across all situations.

We present the evaluation of our model's performance for generation using established metrics commonly employed in machine translation tasks that are BLEU [40, 41], ROUGE-L [42], and METOER [43, 44]. A higher metric value signifies superior model performance, indicating closer proximity between the predicted and ground truth phrases. Notably, these scores were computed exclusively on the test dataset, ensuring that the model had no prior exposure to this data during its training phase. Our findings presented in Table 1 indicate that, individually, input CallTranscriptOnly variant exhibited the highest efficiency among all single data modalities

Table 1: Different Data Modality Variants Performance on Automatic Metrics

Data Modality Variants	BLEU-1	BLEU-2	BLEU-3	ROUGE-L	METOER
CallTranscriptOnly	34.9	24.7	18.5	26.1	17.8
OCRTextOnly	24.8	17.7	11.8	17.1	12.2
SqScreenOnly	16.2	09.4	05.2	12.9	05.7
CallTranscript + SqScreen	42.9	27.3	21.6	32.1	19.2
SqScreen + OCRText	32.7	21.4	14.3	23.7	13.2
CallTranscript + OCRText	49.2	30.7	27.3	32.7	26.5
CallTranscript + OCRText + SqScreen	**58.3**	**46.9**	**34.2**	**41.8**	**35.3**

Table 2: Data Modality Variants Classification Performance on MmVTD Architecture

Call Texts	OCR	Screens	Accuracy
✓	×	×	**77.96**
×	✓	×	**52.47**
×	×	✓	**38.88**
✓	×	✓	**82.27**
×	✓	✓	**54.43**
✓	✓	×	**88.68**
✓	✓	✓	**94.44**

Table 3: Vishing Benchmark Comparison with MmVTD Framework on Different Metrics

Benchmarks	Accuracy(%)	f1-score	AUC
Naive Bayes	63.05	0.59	0.62
KNN	67.71	0.62	0.66
SVM	68.23	0.63	0.66
LSTM	70.61	0.68	0.69
BERT	75.51	0.71	0.72
MmVTD	**77.96**	**0.75**	**0.76**

variants, given its relatively high performance. Significantly, the comprehensive multimodal data outperformed all other data variants across various metrics. This outcome strongly suggests that adopting a multimodal approach substantially improves the capacity to understand pertinent device information, captured during voice calls, and thereby enhances of efficacy of descriptive cautionary alert generation.

The performance system of MmVTD for vishing classification was evaluated using the widely used Accuracy metric. Greater model success is indicated by a larger metric value, which also shows a closer relationship between the predicted and ground truth labels. In particular, the test dataset was the only one used to calculate these ratings. Our results, which are shown in Table 2, demonstrate the superiority of the call transcript modality over the other. Additionally, the combination of modalities performed significantly better than the individual modalities. Similar to text generation evaluation, across all variants, the comprehensive multimodel data performed best, which strongly reflects that multimodality greatly enhances the vishing call categorization power and strongly advocates the adoption of a multimodal strategy.

4.5.2 Vishing Benchmark Comparison. The proposed model for detection of Vishing attacks was subjected to comprehensive evaluation similar to Natarajan et al.[24] which involved a comparison analysis against several benchmark models for call transcripts. The performance metrics in Table 3 reveals that the proposed model's consistency outshines its counterparts, including K-Nearest Neighbour (KNN) [45], Linear Support Vector Machine (SVM)[46], Logistic Regression[47], Long Short-Term Memory (LSTM) [48], Naive Bayes[49] and BERT[50] that uses Transformer Model. When validated against our generated dataset, the proposed solution achieved a commendable efficacy in balancing between correctly identifying transcripts of Vishing conversations while minimizing false positives and negatives. In compassion to LSTM and BERT known for their proficiency in capturing contextual information, the proposed model not only matches but often surpasses their performance, reflecting its competence in handling the unique challenges posed in Vishing classification. The consistent superiority over benchmark models signifies its potential for practical deployment in a variety of contexts where accurate classification is paramount.

4.6 OnDevice Inference

The technique of OnDevice Inference involves placing the pre-trained model on smartphone and perform inference exclusively ondevice without any cloud or server interaction. This approach provides major advantage of privacy on user data that helps us to deal with sensitive information without fear. Though this approach has hardware limitation for inference it can be overcome by device specific model fine-tuning. We utilized Samsung S24 to evaluate performance of our model. The generated model is converted to tflite using tflite converter [51]. The inference implementation is done in kotlin for android using tflite libraries. The model that has best inference time i,e., 250ms (detailed comparative on-device timings presented in Figure 7 and better accuracy is taken as best model to be used on device. The model trained and evaluated on GPU machine has to be packaged in android application to be used on device. Post-training Quantization [52] techniques like Dynamic Range, Full Integer, Float16 and Int8 were experimented to reduce the model size. The model size reduced from 46.08 MB to 27.53 MB and with Float16. Quantization was stopped at this stage to not compromise on model accuracy. Runtime memory usage analyse is done by repeating inference for 10 times and average memory utilization of application is taken from Android Studio Memory Profiler [53]. In addition to model size, quantization helped us to reduce runtime memory by 38%. Since ULFApp is used for experimental purpose, for real-time scenarios, the trained model can be integrated with Audio-To-Text API in android environment with screenshot being captured using virtual buffer in Android. During runtime inference, the data follows same steps as detailed in Section 4.3. Once phone receives incoming call, the application fetches data from all modalities, and make the request in sequence. After obtaining the necessary number of screen frames, the user's touch response initiates the model's inference. This process is repeated continuously with incoming multimodality data fields until a malicious action is anticipated. If malicious action is predicted based on the current sequence, user is alerted using a Alert Dialog [54]. This interrupts the user's sequence of action and ensures to get consent of the user to ensure awareness of his actions. The option is provided to user in the Alert Dialog to proceed or cancel. If user is aware and likes to proceed with the action like copy paste the One Time Password received from bank, click the suspicious url received from unknown source then user will not be interrupted further. Otherwise, User selects to cancel the action and device stop the further actions. This evaluation is repeated until user terminates the phone call.

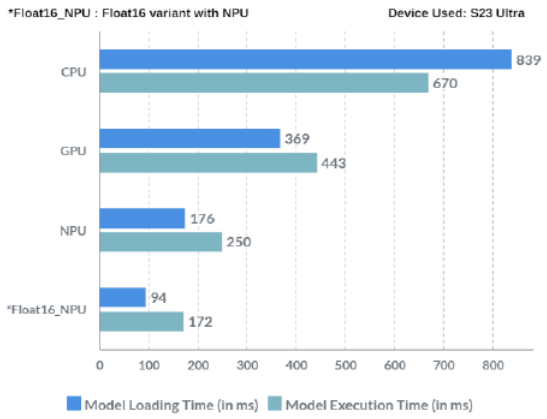

Figure 7: MmVTD Framework On device Timings

5 Conclusion

In this paper, we propose Multimodal Vishing Threat Detection (MmVTD) approach that helps to identify fraudsters deceiving smartphone users to steal information through Vishing. The approach utilises voice call transcript, sequences of screen images and OCR Text to identify user's intention. The MmVTD model trained using usecase specific data collected has proven to be effective in identifying Vishing threats and protecting user by interrupting ones action to warn about the callers trap and consequence. Our exploration on three modalities and evaluation on various combinations of those shows that callTranscript on its own or with all 3 modalities outperform other combinations in classification and alert text generation to warn the user.

References

[1] Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf, and Imtiaz Khan. Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, 3, 2021.

[2] Rana Alabdan. Phishing attacks survey: Types, vectors, and technical approaches. *Future Internet*, 12(10), 2020.

[3] Ankit Kumar Jain and B.B. Gupta. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*, 16(4):527–565, 2022.

[4] Ezer Osei Yeboah-Boateng and Priscilla Mateko Amanor. Phishing, smishing vishing: An assessment of threats against mobile devices. *Journal of Emerging Trends in Computing and Information Sciences*, 5(4):297–307, April 2014.

[5] Shaikh Ashfaq, Pankaj Chandre, Shafi Pathan, Uday Mande, Madhukar Nimbalkar, and Parikshit Mahalle. Defending against vishing attacks: A comprehensive review for prevention and mitigation techniques. In Nihar Ranjan Roy, Sudeep Tanwar, and Usha Batra, editors, *Cyber Security and Digital Forensics*, pages 411–422, Singapore, 2024. Springer Nature Singapore.

[6] Slade E. Griffin and Casey C. Rackley. Vishing. In *Proceedings of the 5th Annual Conference on Information Security Curriculum Development*, InfoSecCD '08, page 33–35, New York, NY, USA, 2008. Association for Computing Machinery.

[7] Carey Jewitt, Jeff Bezemer, and Kay O'Halloran. *Introducing multimodality*. Routledge, 2016.

[8] Elisabetta Adami. Introducing multimodality. *The Oxford handbook of language and society*, pages 451–472, 2016.

[9] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[10] Dhanesh Ramachandram and Graham W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[11] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata. Audio-visual speech recognition using deep learning. *Applied intelligence*, 42:722–737, 2015.

[12] Amit Mehta and Theresa C McLoud. Voice recognition. *Journal of thoracic imaging*, 18(3):178–182, 2003.

[13] Hiranmayi Ranganathan, Shayok Chakraborty, and Sethuraman Panchanathan. Multimodal emotion recognition using deep learning architectures. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016.

[14] Panagiotis Tzirakis, George Trigeorgis, Mihalis A. Nicolaou, Björn W. Schuller, and Stefanos Zafeiriou. End-to-end multimodal emotion recognition using deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1301–1309, 2017.

[15] Sharmeen M Saleem Abdullah Abdullah, Siddeeq Y Ameen Ameen, Mohammed AM Sadeeq, and Subhi Zeebaree. Multimodal emotion recognition using deep learning. *Journal of Applied Science and Technology Trends*, 2(02):52–58, 2021.

[16] Julián N Acosta, Guido J Falcone, Pranav Rajpurkar, and Eric J Topol. Multimodal biomedical ai. *Nature Medicine*, 28(9):1773–1784, 2022.

[17] Muxuan Liang, Zhizhong Li, Ting Chen, and Jianyang Zeng. Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM transactions on computational biology and bioinformatics*, 12(4):928–937, 2014.

[18] Dongdong Sun, Minghui Wang, and Ao Li. A multimodal deep neural network for human breast cancer prognosis prediction by integrating multi-dimensional data. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(3):841–850, 2018.

[19] Maria Koutsikou, Vasilia Christidou, Maria Papadopoulou, and Fotini Bonoti. Interpersonal meaning: Verbal text–image relations in multimodal science texts for young children. *Education Sciences*, 11(5), 2021.

[20] Michail N. Giannakos, Kshitij Sharma, Ilias O. Pappas, Vassilis Kostakos, and Eduardo Velloso. Multimodal data as a means to understand the learning experience. *International Journal of Information Management*, 48:108–119, 2019.

[21] Paulo Blikstein. Multimodal learning analytics. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK '13, page 102–106, New York, NY, USA, 2013. Association for Computing Machinery.

[22] Oren Kedem and Avi Turgeman. System, device, and method of detecting vishing attacks, April 6 2021. US Patent 10,970,394.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[24] Abhiram Natarajan, Anirudh Kannan, Varun Belagali, Vaibhavi N. Pai, Rajashree Shettar, and Poonam Ghuli. Spam detection over call transcript using deep learning. In Kohei Arai, editor, *Proceedings of the Future Technologies Conference (FTC) 2021, Volume 2*, pages 138–150, Cham, 2022. Springer International Publishing.

[25] D.R. Denslin Brabin and Sriramulu Bojjagani. A secure mechanism for prevention of vishing attack in banking system. In *2023 International Conference on Networking and Communications (ICNWC)*, pages 1–5, 2023.

[26] Kwan Choi, Ju-lak Lee, and Yong-tae Chun. Voice phishing fraud and its modus operandi. *Security Journal*, 30:454–466, 2017.

[27] Sumitra Biswal. Real-time intelligent vishing prediction and awareness model (rivpam). In *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pages 1–2, 2021.

[28] Keith S Jones, Miriam E Armstrong, McKenna K Tornblad, and Akbar Siami Namin. How social engineers use persuasion principles during vishing attacks. *Information & Computer Security*, 29(2):314–331, 2021.

[29] Minyoung Lee and Eunil Park. Real-time korean voice phishing detection based on machine learning approaches. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–12, 2021.

[30] P. Xu, X. Zhu, and D. A. Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 45(10):12113–12132, oct 2023.

[31] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2130–2134, 2015.

[32] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 112–118, 2018.

[33] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *International Conference on Computer Vision (ICCV)*, 2021.

[34] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[35] Dominique Brunet, Edward R. Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2012.

[36] GoogleMLKit. Google Text recognition. https://developers.google.com/ml-kit/vision/text-recognition/v2/android, 2023. [Online; accessed 10-11-2023].

[37] Google. Google Speech To Text. https://cloud.google.com/speech-to-text?hl=en.

[38] Daša Munková, Michal Munk, and Martin Vozár. Influence of stopwords removal on sequence patterns identification within comparable corpora. In Vladimir Trajkovik and Misev Anastas, editors, *ICT Innovations 2013*, pages 67–76, Heidelberg, 2014. Springer International Publishing.

[39] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[40] Matt Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

[41] Ehud Reiter. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401, 2018.

[42] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[43] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[44] Alon Lavie and Michael J Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23:105–115, 2009.

[45] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. 2001.

[46] J. R. Méndez, E. L. Iglesias, F. Fdez-Riverola, F. Díaz, and J. M. Corchado. Tokenising, stemming and stopword removal on anti-spam filtering domain. In Roque Marín, Eva Onaindía, Alberto Bugarín, and José Santos, editors, *Current Topics in Artificial Intelligence*, pages 449–458, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[47] Bilge Kagan Dedeturk and Bahriye Akay. Spam filtering using a logistic regression model trained by an artificial bee colony algorithm. *Applied Soft Computing*, 91:106229, 2020.

[48] Gauri Jain, Manisha Sharma, and Basant Agarwal. Spam detection in social media using convolutional and long short term memory neural network. *Annals of Mathematics and Artificial Intelligence*, 85, 01 2019.

[49] Fariska Zakhralativa Ruskanda. Study on the effect of preprocessing methods for spam email detection. *Indonesia Journal on Computing (Indo-JC)*, 4(1):109–118, Mar. 2019.

[50] V. Sri Vinitha, D. Karthika Renuka, and L. Ashok Kumar. Transformer - based attention model for email spam classification. In Vikrant Bhateja, Xin-She Yang, Marta Campos Ferreira, Sandeep Singh Sengar, and Carlos M. Travieso-Gonzalez, editors, *Evolution in Computational Intelligence*, pages 219–233, Singapore, 2023. Springer Nature Singapore.

[51] Google. Android Tflite. https://www.tensorflow.org/lite/models/convert, 2020.

[52] Google. Android Memory Profiler. https://developer.android.com/studio/profile/memory-profiler, 2022.

[53] Google. Android Memory Profiler. https://developer.android.com/studio/profile/memory-profiler, 2020. [Online; accessed 10-11-2023].

[54] Google. Android Dialog. https://developer.android.com/develop/ui/views/components/dialogs, 2017.