INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Adversarial Agent for Synthetic Data Generation for Phishing Detection

FRANCISCO FONSECA FERREIRA CARDOSO
Setembro de 2025

Adversarial Agent for Synthetic Data Generation for Phishing Detection

Francisco Fonseca Ferreira Cardoso
Student No.: 1200860

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Artifical Intelligence Engineering

Supervisor: Dr. Isabel Cecília Correia da Silva Praça Gomes Pereira
Co-Supervisor: Dr. Eva Catarina Gomes Maia

Evaluation Committee:
President:
António Constantino Lopes Martins, Associate Professor, Institute of Engineering, Polytechnic of Porto

Members:
Tiago Manuel Campelos Ferreira Pinto, Associate Professor, University of Trás-os-Montes and Alto Douro
Isabel Cecília Correia da Silva Praça Gomes Pereira, Associate Professor, Institute of Engineering, Polytechnic of Porto

Porto, September 30, 2025

"A man's dream will never die."
— Marshall D. Teach

Abstract

Phishing attacks continue to be a significant security challenge, causing financial and reputational damage to organizations and individuals, with emails being the primary way for these attacks. While modern defenses continue to rely on phishing detection systems, their effectiveness is being challenged by the evolution of these attacks. Attackers are moving from generic emails to highly personalised and context-specific messages, which conventional models struggle to detect. The performance of these systems is mostly limited by the scarcity of specialised, domain-specific training data needed to recognise such threats. This thesis tries to address this gap by introducing CANDACE, a modular framework designed to generate context-aware synthetic email messages to train and improve these detection systems.

The main innovation of CANDACE comes from its dual Knowledge Graph (KG) architecture, which gives the generation process a contextual foundation. The first KG maps external, real-world information about an organization, while the second models its internal structure, such as employees and projects. A Small Language Model (SLM) then uses the information of these KGs, with other important components, such as URL, to generate an email message that is contextually relevant to the domain of the organization.

The contributions of this work include the complete design, end-to-end implementation, and validation of the CANDACE pipeline. A case study in the Public Administration sector presents the framework's ability to produce convincing, context-aware synthetic messages. The findings confirm that contextual grounding is essential for creating better and more focused training data. This research shows the need to move beyond generic emails datasets, to build more resilient detection systems capable of detecting the more sophisticated and personalised phishing attacks.

Keywords: Phishing Detection, Synthetic Data, Email Generation, SLM, Context-Aware

Resumo

Os ataques de phishing continuam a ser um desafio significativo para a segurança, causando prejuízos financeiros e danos à reputação de organizações e indivíduos, sendo os e-mails a principal forma que estes ataques são feitos. Embora as defesas modernas continuem a depender de sistemas de deteção de phishing, a sua eficácia está a ser posta em causa pela evolução destes ataques. Os atacantes estão a passar de e-mails genéricos para mensagens altamente personalizadas e com contexto, que os modelos convencionais têm dificuldade em detetar. O desempenho destes sistemas é limitado principalmente pela escassez de dados de treino especializados e específicos para o domínio, necessários para reconhecer tais ameaças. Esta tese tenta abordar esta lacuna através da introdução do CANDACE, uma *framework* criada para gerar mensagens de e-mail sintéticas com contexto para treinar e melhorar estes sistemas de deteção.

A principal inovação do CANDACE vem da sua arquitetura com dois Grafos de Conhecimento (GC), que dá ao processo de geração uma base contextual. O primeiro GC mapeia informações externas do mundo real sobre uma organização, enquanto o segundo modela a sua estrutura interna, como funcionários e projetos. Um SLM usa as informações desses KGs, com outros componentes importantes, como URL, para gerar uma mensagem de e-mail que seja contextualmente relevante para o domínio da organização.

As contribuições deste trabalho incluem o *design* completo, a implementação *end-to-end* e a validação do pipeline CANDACE. Um caso de estudo no setor da Administração Pública apresenta a capacidade da *framework* de produzir mensagens sintéticas convincentes e com contexto. As conclusões confirmam que o enquadramento contextual é essencial para criar dados de treino melhores e mais focados. Esta investigação demonstra a necessidade de ir além dos conjuntos de dados genéricos de e-mails, para construir sistemas de deteção mais resilientes, capazes de detetar os ataques de phishing mais sofisticados e personalizados.

Palavras-chave: Phishing Detection, Synthetic Data, Email Generation, SLM, Context-Aware

Acknowledgement

I would like to express my sincere gratitude to my supervisors Isabel Praça and Eva Maia, for providing me with this opportunity and guidance, for all the support and advice.

To my family, for the support they provided, that helped me reach where I am right now.

I must also thank my cat, Norte, for the completely unreciprocated support he provided during the long hours or writing and development.

I would also like to thank my friends, for bringing me some peace of mind when it was most needed.

I would also like to extend my thanks to the other members of GECAD, especially the colleagues from the VESTA island.

Finally, I would like to thank myself. This work is the result of my own blood, sweat, and tears.

Contents

List of Figures

List of Tables

List of Acronyms

ACC	Accuracy.
ACM	ACM Digital Library.
AI	Artificial Intelligence.
AMP	Área Metropolitana do Porto.
BART	Bidirectional and Autoregressive Transformer.
BERT	Bidirectional Encoder Representations from Transformers.
BERTScore	Bidirectional Encoder Representations from Transformers Score.
BLEU	Bilingual Evaluation Understudy.
CANDACE	Context-Aware Narrative Development for Adversarial Communication Emails.
CIM	Comunidade Intermunicipal.
CNN	Convolutional Neural Network.
CPA	Código do Procedimento Administrativo.
CRISP-DM	Cross Industry Standard Process for Data Mining.
CTG	Conditional Text Generation.
DL	Deep Learning.
DNN	Deep Neural Network.
DSR	Design Science Research.
EU	European Union.
GAN	Generative Adversarial Network.
GDPR	General Data Protection Regulation.
GPT	Generative Pre-Trained Transformer.
GRU	Gated Recurrent Units.
IEEE	IEEE Xplore Digital Library.
IEFP	Instituto do Emprego e Formação Profissional.
JSON	JavaScript Object Notation.
KB	Knowledge Base.
KG	Knowledge Graph.
LHU	Local Health Units.

LLM	Large Language Model.
LMM	Large Multimodal Model.
LSTM	Long Short-Term Memory.
MCC	Matthew's Correlation Coefficient.
METEOR	Metric for Evaluation of Translation with Explicit ORdering.
ML	Machine Learning.
MLP	Multi Layer Perceptron.
MoE	Mixture of Experts.
MPAG	Meaningful Product Answer Generator.
NLG	Natural Language Generation.
NLL	Negative Log-Likelihood.
NLP	Natural Language Processing.
PARENT	Precision And Recall of Entailed N-grams from the Table.
PERRY	Phishing Email Recognition and Response sYstem.
PII	Personal Identifiable Information.
PLM	Pre-Trained Language Model.
PPL	Perplexity.
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses.
RL	Reinforcement Learning.
RNN	Recurrent Neural Network.
ROUGE	Recall-Oriented Understudy for Gisting Evaluation.
Seq2Seq	Sequence-to-Sequence.
SLM	Small Language Model.
SMOTE	Synthetic Minority Over-sampling Technique.
STCP	Sociedade de Transportes Colectivos do Porto.
TESLA	Task-specific Evaluation with Semantic and Logical Alignment.
TF-IDF	Term Frequency-Inverse Document Frequency.
URL	Uniform Resource Locator.
VAE	Variational Autoencoder.
WoS	Web of Science.

Chapter 1

Introduction

This introductory chapter gives the context and motivation for the developed work in this thesis, while also presenting the problem description, the research questions and the objectives that were elaborated for it.

1.1 Context and Motivation

Phishing has become one of the most prevalent and damaging forms of cybercrime in the digital age. These attacks can have a big impact, harming not only the individuals and organizations that fall victim to them but also the reputations of brands and entities impersonated by the attackers. According to a recent global study by Interisle Consulting [1], phishing incidents increased by 50,000 between May 2023 and April 2024, bringing the total number of reported attacks to just under 1.9 million. The European Union Agency for Cybersecurity (ENISA) Threat Landscape 2024 Report[2] highlighted that the public administration, transportation, banking, business services, and digital infrastructure were the sector more targeted during this period.

The latest UK government Cyber Security Breaches Survey [3] revealed that 43% of businesses identified at least one phishing attack within a 12-month period. Additionally, 35% of businesses reported breaches involving impersonation of organization through emails or online platforms. The financial implications of such cyberattacks are also significant. According to IBM's Cost of a Data Breach Report 2024 [4], the global average cost of a data breach saw a 10% increase over the previous year, reaching $4.88 million, the largest increase observed since the COVID-19 pandemic. In addition to financial losses, phishing also affects the trust in digital system while also presenting operational and legal challenges to organizations.

Email is one of the primary channels for phishing attacks, where fraudulent messages are designed to deceive users. To appear legitimate, these fraudulent emails often impersonate trusted organizations like banks, government agencies, or service providers. Such attacks commonly involve using deceptive tactics to trick users into disclosing their sensitive information, such as passwords, financial details or personal data. According to the APWG Phishing Activity Trends Report for the 1st Quarter of 2025 [5], webmail services were the most targeted sector, accounting for 17.6% of attacks, followed by payment and ecommerce services at 16.3% and 15.3%, respectively.

Most solutions for preventing general phishing attacks rely on either blacklist-based or Machine Learning (ML)-based approaches, which also include Deep Learning (DL) models [6].

While blacklist methods typically achieve higher accuracy, they are ineffective against zero-day phishing attacks since they are not detected or registered in real-time [7]. A significant limitation of ML and DL approaches is their dependence on the data used for their training. This reliance creates a critical vulnerability in current research, as many academic studies continue to use datasets from a decade or more ago, which capture linguistic patterns and attack methods from a different era [8, 9]. This temporal mismatch means that while the models are trained to detect historical threats, they are often ill-equipped to combat the dynamic, sophisticated and context dependent phishing tactics seen today.

1.2 Problem Description

Phishing remains a persistent and evolving threat, causing significant financial and data losses for organizations and individuals alike. While ML and DL models are central to modern phishing detection strategies, their effectiveness often suffers from class imbalance [10, 11]. In most real-world scenarios, benign emails far outnumber malicious ones, creating highly one-sided datasets. This imbalance biases ML models towards the majority (benign) class, reducing their sensitivity to novel and sophisticated phishing attacks, which constitute the critical minority class [10].

To mitigate this, researchers have employed traditional data balancing techniques. These include undersampling, which removes samples from the majority class, and oversampling, which adds minority class samples. However, while simple, undersampling risks discarding valuable information, and oversampling can lead to model overfitting without introducing new, meaningful patterns [12]. More advanced oversampling methods like Synthetic Minority Over-sampling Technique (SMOTE) generate synthetic samples by interpolating between existing minority instances. Yet, SMOTE's effectiveness for complex data like text is limited, with the possibility of creating unrealistic examples that do not capture the semantic nuances of real-world attacks and may even introduce noise that degrades model performance [12, 13]. These techniques are often insufficient for modelling the diverse nature of phishing emails, which rely on complex social engineering tactics and linguistic patterns.

Deep generative models can offer a more powerful solution. Unlike simple interpolation, models such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) learn the underlying data distribution of the minority class to generate entirely new, high-fidelity synthetic samples [14, 15]. By augmenting training sets with diverse and realistic examples, these models can help ML-based detection systems learn more robust decision boundaries, improving their ability to identify threats [10, 16].

While these generative models offer a powerful solution, their practical effectiveness is dependent on overcoming some challenges. For the data to be effective, the synthetic emails must realistically emulate the style, structure, and contextual details of genuine phishing attacks without introducing artifacts or create "hallucinations" that could mislead detection models [17]. The generative process must be carefully controlled to ensure the synthetic data is both diverse and representative of the full spectrum of phishing strategies [18]. The poor quality of synthetic data can risk degrading the performance of the very detection systems it is meant to help.

This thesis addresses this gap by focusing on the usage of advanced generative models to create high-quality, realistic synthetic email datasets. The primary goal is to overcome the limitations of data scarcity and class imbalance, thereby enhancing the resilience and adaptability of ML and DL detection systems against the full range of phishing threats.

1.3 Research Questions and Objectives

To elaborate this thesis, both research questions and objectives were developed to provide a structure for the investigation. The main question driving this study is: *"How can synthetic datasets help improve phishing detection?"* To tackle this broader question, it has been broken down into smaller sub-questions that focus on specific parts of the problem, making it easier to address each aspect in detail. The research questions guiding this thesis are as follows:

- **RQ1:** What are the state-of-the-art methods and techniques for the generation and augmentation of textual data?
- **RQ2:** What are the most critical and replicable features within email messages that characterise email phishing attempts?
- **RQ3:** How can the quality of synthetic data and their impact on performance of phishing detection models be effectively evaluated?

Based on these research questions, a series of objectives were formulated to outline the key steps necessary for achieving the goals of this thesis. They are designed to follow a clear and systematic approach to answering the research questions and building a strong understanding of the topic. The objectives are:

- **OB1:** Explore the state-of-the-art methods and techniques for generating and augmenting text data.
- **OB2:** Identify the critical linguistic and structural features that differentiate phishing emails from legitimate messages.
- **OB3:** Design and implement a generative system capable of producing high-quality, context-aware synthetic email messages.
- **OB4:** Evaluate the quality of the generated emails components by measuring their coherence, fluency, realism and effectiveness in phishing detection.
- **OB5:** Validate the system's effectiveness and applicability in a simulated real-world case study.

With these research questions and objectives, this thesis aims to develop a system that is capable of generating synthetic email messages, that are also contextual relevant, in order to improve the phishing detection of ML and DL models in a specific domain.

1.4 Research Methodology

This thesis adopts the Design Science Research (DSR) methodology as its guiding research framework. DSR is a research paradigm focused on solving real-world problems through the design, development, and evaluation of innovative artifacts [19]. This approach is particularly well-suited for this project, as its primary goal is to create a practical solution, a system for generation of synthetic and contextual relevant email messages.

The DSR process will be structured around the six iterative steps proposed by Peffers et al. [19], which will guide the research process:

1. **Identification of Problem:** The main issue this study focuses on is the lack of high-quality, balanced datasets for training ML and DL models to detect phishing. The

lack of data makes existing detection systems less effective and adaptable. To tackle this, a systematic review will be conducted using the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology to explore the latest generative and augmentative techniques and how they can help create phishing email datasets.

2. **Defining Solution Objectives:** After identifying the problem, the objectives for the proposed solution will be defined. The main objective is to design and use generative models to create high-quality, context-aware synthetic email datasets that help mitigate data scarcity and imbalance while enhancing the performance of phishing detection systems.
3. **Design and Development:** This part of the study involves designing and building the solution. The design process will be guided by the C4+1 architectural model and also inspired by the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology. C4+1 [20] will provide a architectural blueprint for system components, while the inspiration of CRISP-DM [21] will help design a workflow for data preparation, modelling, and evaluation.
4. **Demonstration:** The main contribution of this thesis is an adversarial agent designed to generate synthetic, context-aware phishing emails. To present its utility, it will be validated through a controlled experiment, on a specific domain. The framework will be used to create a synthetic dataset, and its value will be quantified by using this data to train ML models.
5. **Evaluation:** To evaluate this adversarial agent to generate synthetic, context-aware phishing emails, the quality of the generated emails will be assessed using some of the most used metrics for generated text. The main focus will be on the contextual relevance, realism, and coherence of the generated synthetic emails to ensure they have the possibility of being real phishing attempts. Additionally, the impact of the email messages will also be evaluted.
6. **Communication:** The research process, findings, and contributions will be disseminated to the academic and professional communities. The main channels for communication will be this master's thesis and the peer-reviewed articles that will be published in scientific conferences and journals.

Figure 1.1 displays a diagram of the methodology applied.

1.5 Scientific Contributions

This thesis makes several key scientific contributions to the field of phishing detection, focusing on the application of generative models for synthetic data creation. The primary contributions are:

- A literature review of the most used methods and techniques used for text generation and augmentation, the most critical features within email messages for phishing attempts, as well as metrics used to evaluate synthetic data.
- A comparative analysis of two distinct generative techniques, evaluating their effectiveness for generating realistic and contextually relevant phishing emails.

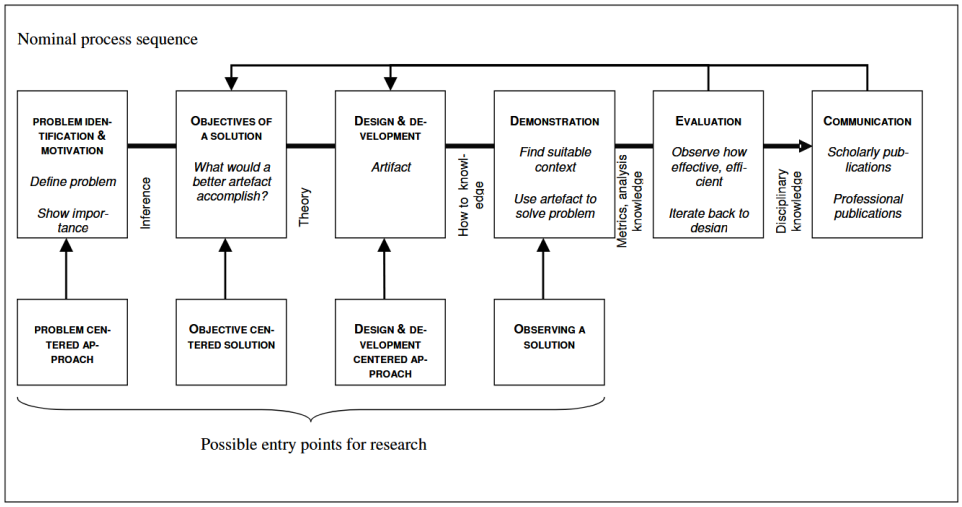

Figure 1.1: Design Science Research Methodology [19]

- An empirical evaluation of how synthetic data augmentation impacts the generalisation performance of ML and DL models, demonstrating its potential to improve robustness against unseen threats.
- The design and implementation of an adversarial agent for synthetic email generation. This agent is a component of Phishing Email Recognition and Response sYstem (PERRY), a robust and scalable system architecture developed within the international VESTA project to provide a framework for real-world applications.

These contributions have been validated and disseminated through the following peer-reviewed scientific publications.

- **Francisco Cardoso**, Eva Maia, Isabel Praça, "Email Augmentation: A Comparison of Fine-tuned Small Language Models", accepted for publication and presented at the *15th International Conference on Advanced Computer Information Technologies (ACIT)*, 2025 [22].
- **Francisco Cardoso**, Eva Maia, Isabel Praça, "Improving Machine Learning Models for URL Phishing Detection using Synthetic Data", accepted for publication and presented at *Computer Security. ESORICS 2025 International Workshops* [23].

In addition to these publications, a related manuscript is currently in preparation.

1.6 Document Structure

This document is organised into several chapters to ensure the flow and ease of reading.

Chapter 1, the current chapter, introduces the context and problem addressed in this thesis. It also outlines the research questions and objectives that serve as a foundation for guiding the work. In addition it also presents the scientific contributions that resulted of this thesis.

Chapter 2 details the systematic review conducted as part of this thesis. It is structured into three sections, each corresponding to one of the research questions. Each section includes a description of the methodology and a presentation of the findings. Following these sections,

the chapter concludes with a dedicated section that synthesises the insights gained from the review and provides overarching remarks.

Chapter 3 discusses the ethical and transparency principles and data handling standards that guided this work, details the procedures followed to ensure adherence to them as well as presents on what Generative Artificial Intelligence (AI) was used.

Chapter 4 shifts focus to the proposed system. It provides a detailed description of the individual components, their architecture, and the data used in their development.

Chapter 5 details the experimental evaluation of two specific modules from the framework, with each assessed using its own set of specific metrics. These experiments provide insights into the challenges of data augmentation for datasets, as well as model differences.

Chapter 6 details a specific case study in which the system was applied. The case study showcases the practical advantages and contextual relevance of the developed system.

Finally, Chapter 7 concludes the thesis with a comprehensive evaluation of the work. It highlights the main contributions and discusses the implications of the findings. Additionally, this chapter suggests future research directions to further improve the system.

Chapter 2

State-of-the-art

This chapter provides a comprehensive literature review aimed at addressing the research questions presented in the previous chapter. Each section outlines the methodology employed and presents the corresponding findings for each research question.

2.1 Text Data Generation and Augmentation

This section addresses RQ1: "*What are the state-of-the-art methods and techniques for the generation and augmentation of textual data?*". It aims to explore tools, methods, and approaches used to generate and enhance textual data, outlining their uses, advantages, and limitations.

2.1.1 Methodology

The investigation addressing RQ1 adopted a systematic review approach guided by the PRISMA framework [24]. This process involved an initial screening of titles and abstracts, followed by a rigorous full-text review of the shortlisted articles to determine their final eligibility for inclusion.

The search strategy was designed to capture an overview of the field. To achieve this, keywords like *text data augmentation* and *text data generation* were paired with secondary terms such as *survey* and *review*. This approach was chosen to identify relevant publications that synthesise and compare a wide array of methods, providing a broader understanding than could be achieved by analysing individual, highly specific research papers.

Table 2.1 provides a summary of the search keywords categorised by domain. Furthermore, terms related to specific applications, such as "context awareness", were deliberately excluded from this initial search. This decision was made to ensure the review remained broad, capturing all relevant text generation and augmentation methods, rather than prematurely narrowing the scope to only those techniques designed for context-dependent tasks.

Table 2.1: Search Keywords for RQ1

Domain	Keywords
Text Data Augmentation	("text data augmentation" OR "text data generation" OR "syntethic text data generation" OR "text generation")
Reviews	("survey" OR "review")

The literature search was conducted using three prominent bibliographic databases: the IEEE Xplore Digital Library (IEEE) [25], the ACM Digital Library (ACM) [26], and Web of Science (WoS) [27]. These databases were selected for their comprehensive coverage of computer science, engineering, and related disciplines, ensuring a thorough survey of the relevant research landscape. The IEEE is a resource for research in electrical engineering, electronics, computer science, and related fields. The ACM is a repository for computing and information technology research. Meanwhile, WoS is a database that contains high-quality scholarly journals, conference proceedings, and books across a diverse range of disciplines, including science, engineering, social sciences, arts, and humanities.

The search query was restricted to English publications from 2019 onward to focus on the most current state-of-the-art techniques. To enhance the precision and relevance of the retrieved articles, the keyword search was targeted at the title, abstract, and author keywords of the publications. In addition to the database queries, backward snowballing, which is a process of reviewing the reference of included articles, was employed to identify other significant studies that may have been missed in the initial search.

To ensure the relevance and quality of the results, inclusion and exclusion criteria were systematically applied. Publications meeting the inclusion criteria were selected to be included in the review. The specific inclusion and exclusion criteria employed in this process are outlined in Table 2.2.

Table 2.2: Inclusion and Exclusion Criteria for RQ1

Inclusion Criteria	Exclusion Criteria
IC1: Surveys, reviews, or introduces techniques for text generation/ augmentation IC2: Provides a clear contribution to the state of the art	EC1: Articles published before 2019 EC2: Duplicate articles from different databases EC3: Articles not published in English EC4: Grey literature (theses, book chapters, editorials) and workshop proceedings

2.1.2 Findings and Discussion

A total of 275 records were initially retrieved for RQ1 by applying the search query to the selected databases. After filtering for publications from 2019 onward, 149 records remained. From these, 23 duplicates were removed, and an additional 74 articles were excluded during screening based on title, abstract, and keywords, leaving 52 records for full-text assessment. Applying exclusion criteria (EC3, EC4) removed 16 records, resulting in a final selection of 36 publications for the review. The PRISMA flowchart is presented in Figure 2.1.

The following subsections delve into the reviewed literature, which is organized into three distinct thematic areas. Before discussing these categories, it is important to distinguish between two fundamental concepts: text generation and text augmentation. Although distinct, they are discussed in conjunction because they both serve the critical goal of creating synthetic data to overcome the limitations of existing training datasets. Text generation focuses on creating entirely new, human-like text from an input, such as a prompt or a set of data. In contrast, text augmentation aims to expand a dataset by creating modified versions of existing text [28]. The primary objective of generation is novel content creation, whereas

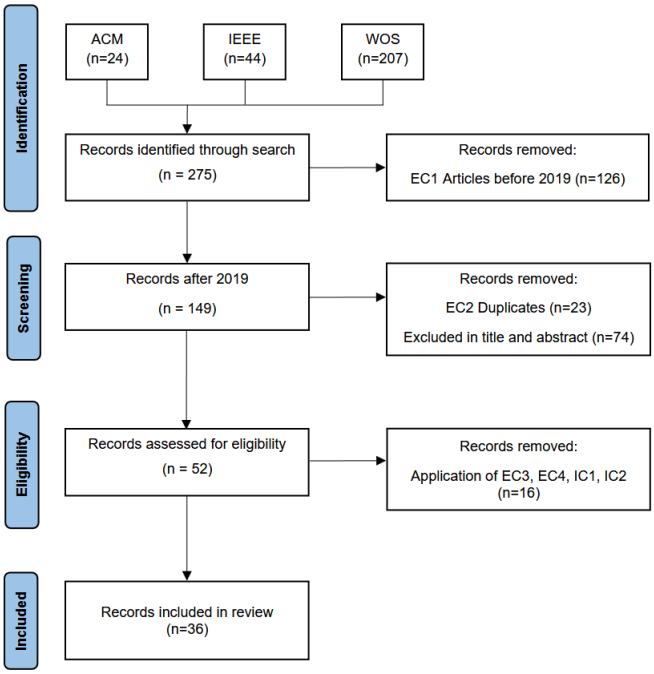

Figure 2.1: PRISMA Search Process for RQ1

augmentation is a technique used to increase the diversity of training data to improve ML model robustness and mitigate overfitting [12].

With this distinction in mind, the literature review is structured as follows. It begins with traditional techniques, which generally operate based on heuristic rules or statistical properties of language, manipulating existing text at a lexical or syntactic level without a deep semantic understanding [29]. Following this, the discussion shifts to deep learning-based approaches, which use neural network architectures trained on vast amounts of data. These models are able to learn intricate patterns and long-range dependencies in language, enabling the generation of more fluent, coherent, and contextually novel text [30]. Finally, the last subsection addresses context-aware and conditional text generation. This area focuses on techniques that can help guide the generation according to specific constraints, ensuring the output is not only of high quality but also highly relevant to a given input, desired style, or other predefined attributes [31, 32].

2.1.2.1 Traditional Methods for Text Generation

Traditional approaches are characterised by their reliance on explicit rules, statistical patterns, and structured data [29]. While now largely unused in many applications, they offer a high degree of control and interpretability, making them relevant for specific, constrained tasks [33]. The methods can be broadly grouped into template-based, statistical, and rule-based systems for generation.

Template-based methods is one of the most simple approaches to text generation, which involves creating static text skeletons with placeholders that are dynamically filled with data. This method ensures grammatical correctness and output consistency, making it highly effective for repetitive tasks where the structure of the output is mostly fixed. The study [34] provides a clear application of this, where they developed a system to generate

summaries for a repository of academic works. By defining a template such as "*... (name) et al. (year) ... proposed a new suggestion by using the... (name of the method) method*", the system could create structured and predictable textual content. This work presents a strength of the template-based approach, which is its ability to generate reliable output from structured information. The quality of such outputs is often confirmed through human evaluation, given their predictable nature.

Besides rigid templates, statistical methods introduced a probabilistic dimension to text generation. As detailed in the survey by [35], these techniques marked an evolution by learning word sequence probabilities from large text corpora. The foundational models in this category are n-gram and Markov models, which calculate the probability of a word occurring based on the preceding sequence of $n-1$ words. The fluency of these models was often measured using Perplexity (PPL). These early Natural Language Generation (NLG) systems could generate more varied text than templates and were successfully applied in domains with predictable linguistic patterns, such as weather forecasting [33]. However, the study [35] also highlights their main limitation, a failure to capture long-range dependencies and semantic context, which constrained their effectiveness in more complex, open-domain generation tasks.

Rule-based methods, in contrast to the probabilistic nature of statistical models, generate text by adhering to a set of hand-crafted linguistic rules [33]. These systems rely on explicit grammars and heuristics defined by experts to construct sentences, offering a high degree of control and precision for specific, narrow domains. The survey by [35] discusses techniques that exemplify this approach, such as generating text summaries using rules based on sentence position or term frequency, where the quality of the output could be checked with metrics like Recall-Oriented Understudy for Gisting Evaluation (ROUGE). However, the core limitation of these methods, which is widely acknowledged in the field, is their rigidity and inability to scale [33, 35]. For this reason, while foundational, rule-based systems have been largely superseded by more flexible and adaptive deep learning approaches in the pursuit of state-of-the-art text generation [33].

2.1.2.2 Deep Learning Approaches for Text Generation and Augmentation

Deep learning has revolutionised text generation and augmentation by enabling models to learn complex patterns from large corpora. These methods use advanced neural architectures, such as transformers, to capture contextual and semantic nuances, enabling the generation of coherent, diverse, and human-like text.

The move to more modern text generation started with Recurrent Neural Networks (RNNs), which were models made to handle sequential data like text [36, 37]. However, these early RNNs struggled to learn from long sentences or paragraphs because of an issue known as the vanishing gradient problem. This was a challenge that many studies pointed out [38, 39]. It led to the creation of better models, mainly Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) networks. These newer models used "gates" to control what information to keep or forget, which let them connect words and ideas over longer stretches of text. This made them much better at creating text that made sense, and they became the standard deep learning models for text generation for a time [40, 41].

LSTMs proved to be useful for many different text generation and data augmentation tasks, with their outputs typically assessed using metrics like Bilingual Evaluation Understudy (BLEU) and ROUGE. For example, a study by [40] developed an LSTM-based framework

specifically to generate both abstractive and extractive summaries from marketing articles, noting that the model's success in creating relevant content was highly dependent on the diversity of the training data. In the domain of data augmentation, the authors of [42] proposed a novel method to combat data sparsity in recommendation systems by generating synthetic user reviews. Their model, an LSTM network enhanced with an attention mechanism, was able to produce high-fidelity reviews that could successfully augment training data, with the attention layer being critical for ensuring the generated text was personalized and coherent. Showcasing a different application, the work by [43] integrated an LSTM as the text generation component in their Meaningful Product Answer Generator (MPAG) model. In this architecture, the LSTM decoder's role was to synthesise information from multiple sources (like product parameters and user reviews) to generate diverse and informative answers to complex product-related questions.

Researchers also worked on improving these core models. For instance, a comparative analysis was conducted by [39] on models like Bidirectional-LSTMs (Bi-LSTMs) and Bidirectional-GRUs (Bi-GRUs) by evaluating their ability to generate text at the character level. By training them on literary datasets like "Alice in Wonderland" [44] and "Hansel and Gretel" [45], they were able to use ROUGE and BLEU to directly compare the models' efficiency and their skill at learning and recreating a specific author's writing style. Even with these improvements, some problems remained, such as the models overfitting to the training data or the difficulty of using them for languages with less available data [38].

Even though LSTMs were a big improvement, they are no longer considered the state-of-the-art for most text generation tasks. A core reason is their sequential nature, because they process text word-by-word and are computationally intensive. Furthermore, they struggle to capture the global context of a long document as effectively as newer architectures [46]. In a detailed review of the field's history, the authors of [37] describe RNNs and LSTMs as important early steps that excelled at handling sequences, but their inherent problems led directly to the creation of the encoder-decoder architecture. This idea is widely supported, with multiple surveys pointing out that even advanced Sequence-to-Sequence (Seq2Seq) models using LSTMs still had trouble with long-term dependencies [41, 47]. This specific limitation was the direct motivation for the creation of the Transformer architecture, which has since become a dominant technology in the field [37, 41].

The introduction of the Transformer architecture represented a fundamental shift in text generation, directly addressing the critical limitations of prior recurrent models [41, 48]. Unlike RNNs and LSTMs, which process text sequentially, Transformers use a self-attention mechanism that allows them to weigh the importance of all words in the input sequence simultaneously [37]. This non-sequential processing not only resolved the vanishing gradient problem and enabled the capture of complex, long-range dependencies but also allowed for massive parallelisation, which was crucial for training the next generation of models [35, 46].

This architectural innovation paved the way for the era of large-scale Pre-Trained Language Models (PLMs), establishing Transformers as the foundation for state-of-the-art text generation. Models such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-Trained Transformer (GPT) showed that by pre-training on vast amounts of text, a Transformer-based model could acquire a deep, contextual understanding of language that could be fine-tuned for a wide range of downstream tasks, producing text with unprecedented fluency and coherence [46, 48]. This flexibility also enables the creation of highly specialised and efficient models. For example, the authors of [49] developed a decoder-only Transformer tailored for domain-specific tasks. By pre-training on a more focused dataset,

they created a model that could be effectively fine-tuned for e-commerce applications like title and review summarisation, presenting how the Transformer architecture can be adapted for more targeted use cases without requiring massive, general-purpose corpora.

Beyond foundational improvements, Transformers have enabled significant progress in specialised and controlled generation tasks. To address the need for fine-grained control, a study by [50] proposed the Hierarchical Template Transformer, a model designed for sentiment-controllable text generation. Their two-phase framework first generates sentiment-specific phrases based on given aspect terms and then uses a Transformer to assemble these phrases into a coherent review, achieving better performance on the FSCG-80 corpus, derived from Yelp reviews, than the existing methods for fine-grained sentiment controllable generation tasks, in fluency and sentiment accuracy. Furthermore, Transformers have proven highly effective in data-to-text generation, which involves converting structured data into natural language. The survey by [51] details how encoder-decoder Transformer models are used to interpret structured inputs like tables or graphs and generate descriptive text. For this task, where fidelity to the source data is often evaluated, specialised metrics like Precision And Recall of Entailed N-grams from the Table (PARENT) are used. These applications show that Transformers are not just generating fluent text, but are also being adapted to achieve fine-grained control and ensure factual consistency in complex tasks.

However, despite this progress in specialised applications, better controllability remains one of the most prominent challenges. The difficulty of ensuring that the generated text adheres to specific attributes, styles, or factual constraints, remains a difficult task due to the limited interpretability of these large models [41, 52]. A closely related and critical problem is "hallucination", where models generate fluent and convincing-sounding text that is factually incorrect or nonsensical [17, 51]. Furthermore, the large scale of modern PLMs presents immense practical challenges, including prohibitive computational costs for training and fine-tuning, as well as concerns regarding data privacy and model bias [47].

Besides Transformers, GANs offer a different way to generate text. The main idea behind a GAN is to have two neural networks, a "generator" and a "discriminator", compete against each other. The generator's job is to create new text, while the discriminator's job is to tell the difference between the generator's fake text and real, human-written text. Through this process, called "adversarial training", the generator gets better and better at producing text that is so realistic it can fool the discriminator [53]. A significant challenge in text generation is the application of GANs, due to the discrete nature of text. The usage of separate words makes it difficult to backpropagate training signals from the discriminator to the generator, unlike with continuous data such as images [14]. This has led researchers to create several clever solutions to make this training method work for text.

The most popular solution has been to use Reinforcement Learning (RL) to get around the problem. In this setup, the generator acts like a player in a game, and its policy is to choose the best next word for a sentence [54]. The score from the discriminator is then used as a reward to guide the generator. The SeqGAN model was the first to successfully use this method, employing a technique called Monte Carlo search to give feedback on partially finished sentences [55]. The quality of its output was assessed using BLEU and a specialised metric, Negative Log-Likelihood (NLL), to evaluate model fit [55, 56]. This RL-based approach was built upon in later models like VGAN [57]. Other ways to solve the problem with discrete words includes using the Gumbel-Softmax distribution, which creates a "softer" version of the text data that allows training signals to pass through, or by designing different training goals, like in MaliGAN [58], to make the training more stable [14].

These advanced techniques have allowed GANs to be used for complex and controlled generation tasks. For example, research by Nezhad et al. [59] successfully used the CATGAN model to generate persuasive text designed to match a user's specific personality. In another case, a new GAN system was created by [60] where a Bi-LSTM-based generator created text with specific sentiments. The discriminator would then check the generated text for errors, allowing the generator to be slowly improved until it could produce fluent and emotionally accurate sentences, performing better than older models like Seq2Seq [61]. Other models like RankGAN [62] and LeakGAN [63] were also developed to tackle the ongoing challenges of making the generated text more diverse and meaningful [53].

Despite these clever solutions, GANs and RL-based methods are not the most common choice for text generation today. They still face major challenges, including unstable training and a problem known as "mode collapse", where the generator gets stuck and produces only a very limited variety of sentences [38, 48]. This makes them a powerful tool for tasks requiring high diversity or for research exploring novel training objectives. However, due to these persistent challenges, Transformer-based models remain the more reliable and widely adopted state-of-the-art approach for most text generation applications.

VAEs represent another significant approach in deep learning for text generation, offering a probabilistic method for modelling data. A VAE works by using an encoder to compress input text into a latent space, a compressed, numerical representation of the text's semantic features, and a decoder to reconstruct the text from this latent space [53]. This unsupervised framework is powerful for generating diverse text, but it faces a critical and well-known challenge known as "posterior collapse". This occurs when the decoder learns to ignore the latent space and instead generates generic, safe sentences on its own, making the model ineffective [53]. To overcome this, researchers have developed several techniques, such as modifying the training process with KL-annealing and word dropout, which encourage the decoder to rely more on the latent code [64]. A more advanced solution is to structure the latent space itself. For example, the Topic-Guided VAE (TGVAE) imposes a Gaussian mixture model on the latent space, where each component represents a distinct topic [65]. By doing so, the TGVAE provides stronger guidance to the decoder, not only preventing posterior collapse but also enabling the generation of coherent sentences that are explicitly aligned with a desired topic. The quality of these generated texts is typically evaluated using metrics like PPL to measure fluency and BLEU to compare against reference sentences. This shows how architectural innovations are addressing the core weaknesses of VAEs, making them a more robust and controllable method for text generation.

A more recent and highly promising state-of-the-art technique is text diffusion, which generates text in a non-autoregressive manner. These models work by starting with a sequence of random noise, either discrete tokens like "[MASK]" or continuous Gaussian noise, and then progressively refining it into a coherent sentence over a series of steps [66]. A denoising network, typically a Transformer, is trained to reverse this corruption process, learning to recover clean text from its noisy versions. This approach has shown great potential for various generation tasks, including unconstrained, conditioned, and multi-modal generation [67].

This core diffusion technique has been successfully adapted and extended for a range of specialised applications. For Seq2Seq tasks, DiffuSeq introduced a partial noising strategy to preserve the meaning of the source sentence while generating the target [68], a concept that was later built upon by models like SeqDiffuSeq, which added improvements such as adaptive noise schedules [69]. For extractive summarisation, DiffuSum adapted the diffusion

process by adding specific matching and contrastive loss functions to better identify salient sentences [70]. To improve fine-grained control and overall quality, other models have integrated different techniques. For example, Diffusion-LM employed a continuous diffusion process for better controllability [71], while DiffusionBERT combined the powerful pre-trained representations of BERT with a discrete diffusion model to enhance performance in unconditional text generation [72]. The performance of these models was measured with a suite of metrics, including PPL and BLEU. More advanced, flexible systems like Self-conditioned Embedding Diffusion (SED) [73] and Semi-autoregressive Simplex-based Diffusion Language Model (SSD-LM) [74] have shown the potential for multi-task diffusion, capable of handling both conditioned and unconstrained generation within a single, modular framework.

The development of PLMs marks a big moment in text generation, establishing the current state-of-the-art. By pre-training on enormous text corpora, models like BERT, GPT, and T5 learn a deep and nuanced understanding of language, which can then be adapted to specific tasks through a process called fine-tuning [37]. This "pre-train and fine-tune" paradigm has led to significant improvements in the fluency, coherence, and contextual awareness of generated text across a wide range of applications [35]. The architectural design of these models is very important to their function. As detailed in the survey by [47], PLMs generally fall into three categories: encoder-only models like BERT [75], which excel at understanding tasks; decoder-only models like GPT [76], which are designed for autoregressive text generation; and encoder-decoder models like T5 [77] and Bidirectional and Autoregressive Transformer (BART) [78], which are well-suited for Seq2Seq tasks such as summarisation and translation.

The power of PLMs lies in their adaptability to highly specialised and complex generation tasks. For instance, a "document sketching" method was developed by [79] that uses PLMs to generate draft templates from similar documents, employing advanced techniques like Mixture of Experts (MoE) and RL to ensure the final output is well-structured and relevant. PLMs have also been central to achieving fine-grained control over generated content. A study by [80] demonstrated this by extending a GPT-2 model with a custom module to generate product reviews with precise, aspect-level sentiment control. The real-world application of these models is also evident in highly sensitive domains. In a neuro-rehabilitation setting, the study by [81] used the RosaeNLG platform, built on these principles, to gener ate personalised and context-aware text for communication between care robots and patients, dynamically adapting the language to the patient's progress.

Large Language Models (LLMs) represent the current state-of-the-art, and are essentially PLM architectures that have been scaled up to billions or even trillions of parameters. Models such as GPT-3 and GPT-4, trained on internet-scale data, can produce text that is often indistinguishable from human writing. An important evolution in this area is the shift to Large Multimodal Models (LMMs), which extend the capabilities of LLMs beyond text to integrate and process other data types like images and audio, enabling more complex applications like visual reasoning [82]. The impact of this technology is already being seen in specialised fields like healthcare, where LLMs are being used to generate synthetic medical data for training other AI models, summarise complex research papers, and improve clinical question-answering systems through AI Generated Content [83].

As a direct result of their complexity, the effectiveness of these advanced model architectures is critically dependent on the quality and scale of their training data. To meet this demand and further improve performance, Text Data Augmentation has also become a technique that helps mitigates this issue. Modern methods of data augmentation can be grouped into

several families, moving beyond simple transformations. The first category includes symbolic (rule-based) techniques, such as synonym replacement, random word insertion, and syntactic reordering [12]. A second, more advanced category is generative augmentation. This involves using sophisticated models to create entirely new data, with common examples being back-translation or the use of conditional PLMs to synthesise label-preserving sentences. A third distinct approach is interpolation-based augmentation, notably Mixup, which creates new training instances by combining the hidden vector representations of two or more existing examples. Finally, graph-based methods where text is converted into a graph structure and augmentations are performed by modifying its nodes or edges. The practical impact of all these techniques is significant, particularly in low-resource scenarios. For example, the study by [13] successfully applied augmentation to improve sentiment classification for Spanish text, while the paper by [84] highlights its importance for addressing data scarcity in under-resourced languages like Cantonese.

2.1.2.3 Context-Aware and Conditional Text Generation Techniques

To move beyond generating generic text, a area of state-of-the-art research focuses on making generation context-aware and controllable. This involves grounding the generated text in external knowledge and giving developers explicit control over the output's characteristics.

One major approach is knowledge-enhanced generation, which tries to make text more factual and relevant by connecting it to structured information. The survey by Yu et al. [32] explains that while powerful, standard PLMs like BERT and T5 often struggle to use structured world knowledge. To solve this, techniques like attention mechanisms, copy and pointing networks, and graph neural networks are used to feed external information from knowledge bases or knowledge graphs directly into the generation process. This helps to reduce factual errors and produce text that is more informed and accurate.

Building on this, Conditional Text Generation (CTG) focuses on controlling stylistic and semantic attributes of the output. As reviewed by Guo et al. [31], the goal is to direct the generation process using specific conditions, such as a desired emotion, topic, or level of personalisation. The survey on controllable generation by Zhang et al. [52] details the primary methods for achieving this with modern Transformer models. These methods include fine-tuning a PLM using techniques like prompt-based tuning and RL-inspired approaches, or using post-processing methods to guide the output during the decoding stage. These techniques are important for applications like dialogue systems, storytelling, and removing biases from models to ensure the generated text meets specific requirements.

A practical application of these principles is seen in the CoRe framework, which was designed to automatically generate replies to user app reviews [85]. The system used a Bi-GRU-based encoder-decoder model that incorporated contextual knowledge, such as official app descriptions and other similar user reviews, to produce relevant and helpful responses. To ensure accuracy, the model also used a pointer network, which allowed it to directly copy important details like usernames or specific bug descriptions from the input review into the generated reply, demonstrating how these techniques can be combined to build effective, real-world systems.

2.1.3 Summary of Text Generation and Augmentation Techniques

To synthesise the findings from this chapter, Table 2.3 provides a comprehensive overview of the primary text generation and augmentation techniques discussed. The table is structured

to follow the historical and technological evolution of the field, starting with traditional methods and progressing to the current state-of-the-art. It highlights the main advantages and main challenges of each approach, serving as a consolidated reference. It is important to note the relationship between architectures and paradigms. For instance, the Transformer is the fundamental architecture that enabled the development of the PLM paradigm.

The techniques summarised in the table reveal a clear distinction in the field of text generation. Traditional methods while offering a higher degree of control and interpretability, are limited in their scale and linguistic flexibility. In contrast, DL-based approaches, especially those built on the Transformer architecture, provide unparalleled performance in generating fluent, coherent, and contextually-aware text. However, this performance comes at the cost of high computational demand, a need for vast datasets, and challenges related to controllability and factual accuracy. The ongoing research into context-aware and knowledge-grounded models aims to bridge this gap by combining the generative power of these models with the precision required for domain-specific applications.

Evaluating the quality of generated text is a complex challenge in itself, and selecting the appropriate metric is as crucial as designing the model architecture. No single metric can capture all desirable qualities, such as fluency, coherence, diversity, and factual accuracy. Therefore, researchers rely on a diverse set of evaluation methods, each with its own strengths and weaknesses. The following is a consolidated overview of the most common metrics that appear throughout the literature reviewed in this chapter. Table 2.4 provides a summary of the techniques associated with each metric, offering a high-level view of their application:

- BLEU: This metric [89] calculates the similarity of a generated text to a reference text based on the weighted average of matching n-gram phrases. However, BLEU primarily measures precision and does not consider recall, which can lead to high scores even for incomplete outputs.
- ROUGE: This metric [90] complements BLEU by focusing on recall. It measures the similarity between generated and reference texts by counting overlapping n-grams (ROUGE-N) or the longest common subsequence (ROUGE-L).
- PPL: Perplexity [91] evaluates the quality of a language model by measuring its uncertainty in predicting the next word. A lower perplexity score indicates better performance and is often used as a proxy for the fluency of the generated text.
- Metric for Evaluation of Translation with Explicit ORdering (METEOR): This metric [92] improves upon BLEU by incorporating synonym matching and stemming, providing a more nuanced assessment of fluency and semantic alignment.
- NLL: Negative Log-Likelihood was used in the context of SeqGAN [55] to measure how well the generated data is explained by an oracle language model. Lower NLL values indicate better model performance.
- Bidirectional Encoder Representations from Transformers Score (BERTScore): This metric [93] leverages pre-trained contextual embeddings from BERT to compute the similarity between generated and reference sentences based on cosine similarity, capturing deeper semantic relationships.
- Distinct: This metric [94] evaluates the diversity of generated text by calculating the ratio of unique n-grams to the total number of n-grams, which is useful for detecting and avoiding repetitive outputs.

Table 2.3: Summary of Text Generation Techniques

Technique	Type	Advantages	Challenges	Citations
Template-Based Methods	Generation	Controlled and consistent outputs	Limited scalability	[34]
Statistical Methods (e.g., n-grams)	Generation	Simple, effective for specific domains	Handle diverse contexts	[35]
Rule-Based Methods	Generation / Augmentation	High precision, tailored outputs	Manual configuration, limited scalability	[12, 35]
RNNs / LSTMs	Generation	Effective for sequential tasks, memory retention	Long-term dependencies	[36–38, 40–43, 46]
Transformers	Generation	Scalability, coherence in text generation	Data-intensive, computational cost	[35, 37, 41, 46, 48–51]
GANs	Generation / Augmentation	High-quality outputs, diversity	Mode collapse, challenging for discrete data	[14, 38, 48, 53–63, 86–88]
VAEs	Generation	Diversity in outputs, probabilistic modelling	Posterior collapse	[53, 65]
Diffusion Models	Generation	Fine-grained control, diversity	Emerging technique	[66–74]
PLM	Generation / Augmentation	High fluency, coherence, context-awareness	Computational cost, domain adaptation challenges	[35, 37, 47, 79–81]
Knowledge-Grounded Models	Generation	Factually grounded, domain-specific outputs	Complexity in integration, knowledge maintenance	[32]
Context-Aware / Conditional Generation	Generation	Personalized, relevant outputs	Requires accurate conditioning, resource-intensive	[31, 52, 85]
Graph-Structured Augmentation	Augmentation	Adds structural data, enables label-preserving transformations	Computational overhead	[12]
MixUp Augmentation	Augmentation	Connects distant points, reduces overfitting	Blending may create invalid labels	[12]
Feature Space Augmentation	Augmentation	Adds diversity in latent space	Can lead to overfitting if noise is excessive	[12]
Back-Translation Augmentation	Augmentation	Improves model diversity	Requires quality machine translation models	[12]
Style Augmentation	Augmentation	Focuses models on semantics, reduces overfitting	May blur original style	[12]

- Accuracy (ACC): Accuracy is a standard metric for classification tasks and structured outputs, measuring the proportion of correct predictions.
- F1-Score: This score balances precision and recall, providing a single, robust measure of performance, especially for structured augmentation tasks or imbalanced classification.
- Novelty: This metric [95] measures how different the generated sentences are from the training corpus, often calculated using the Jaccard index to compare sentence overlap.
- Diversity: This metric [95] assesses whether a generator produces a variety of sentences rather than repetitive outputs. High diversity scores suggest the model can produce engaging text.
- Task-specific Evaluation with Semantic and Logical Alignment (TESLA): This is a syntax-based metric that measures how well the grammatical structure of the generated text aligns with that of reference texts [96].
- PARENT: This metric [97] is designed for data-to-text generation and evaluates both the fidelity of the generated text to the source data and its overall fluency.
- MoverScore: This metric [98] uses contextual embeddings and Earth Mover's Distance to evaluate semantic overlap between generated and reference texts.
- Human Evaluation: This remains the gold standard for qualitative assessment, relying on human judgements of fluency, coherence, relevance, and intelligibility to validate machine-generated text against human expectations.

2.2 Critical Features in Emails for Phishing Detection

To effectively generate synthetic phishing emails, it is first required to understand their fundamental components. This section, therefore, addresses RQ2: "*What are the most critical and replicable features within email messages that characterise email phishing attempts?*". It identifies and categorises the most critical features that are instrumental for phishing detection.

2.2.1 Methodology

To address RQ2, the investigation's focus shifted from a broad survey of methods to a targeted systematic review of primary research studies. The goal was to identify and catalogue concrete email features used directly in phishing detection systems. This required a methodology that prioritised original research articles over secondary summaries. The review process followed the PRISMA framework guidelines, beginning with an initial screening of titles and abstracts, followed by a rigorous full-text review to determine final eligibility.

The search strategy was designed to intersect two domains: the specific components of an email and the application to phishing detection. To achieve this, keywords related to email components such as *email metadata* and *email content* were combined with terms related to the application, such as *phishing detection* and *phishing attacks*. This approach was chosen to to retrieve articles that were not only about phishing in general, but also discussed the main features and data used in detection systems. Table 2.5 provides a complete summary of the search keywords used.

Table 2.4: Summary of Metrics used for Text Generation and Augmentation

Metric	**Examples of Techniques**
BLEU	Statistical Methods, RNNs / LSTMs, Transformers, GANs, VAEs, Diffusion Models, PLM, Knowledge-Grounded Models, Context-Aware / Conditional Generation
ROUGE	Statistical Methods, RNNs / LSTMs, Transformers, VAEs, Diffusion Models, PLM, Knowledge-Grounded Models, Context-Aware / Conditional Generation
PPL	RNNs / LSTMs, Transformers, GANs, VAEs, Diffusion Models, PLM, Knowledge-Grounded Models, Context-Aware / Conditional Generation
METEOR	Statistical Methods, RNNs / LSTMs, Transformers, PLM, Context-Aware / Conditional Generation
Distinct	Transformers, GANs, Diffusion Models, PLM, Knowledge-Grounded Models, Context-Aware / Conditional Generation
NLL	GANs, VAEs, Diffusion Models, PLM
Diversity	GANs
Novelty	GANs
BERTScore	Diffusion Models
PARENT	Transformers
TESLA	Context-Aware / Conditional Generation
ACC	Rule-Based Methods, Feature Space, MixUp Augmentation, Back-Translation, Style Augmentation, Context-Aware / Conditional Generation
F1-Score	Rule-Based Methods, Graph-Structured
Human Eval.	Template-Based Methods, RNNs / LSTMs, Context-Aware / Conditional Generation, GANs
MoverScore	Statistical Methods

The same databases utilised for the RQ1 search, IEEE, ACM, and WoS, were also employed for the search addressing RQ2. The queries targeted publications published from 2019 onward and limited to those available in English.

As with the previous research question, the literature search was conducted using the same three bibliographic databases: the IEEE, the ACM, and WoS. The search query was restricted to English publications from 2019 onward to focus on the most current and relevant research. To enhance precision, the keyword search was specifically targeted at the title, abstract, and author keywords of the publications.

To ensure the review was focused on high-quality research, a set of inclusion and exclusion criteria was applied. The inclusion criteria were designed to capture primary research. Articles had to propose or validate specific email features (IC1) and present an empirical study or system implementation (IC2). In parallel, the exclusion criteria filtered out non-relevant

Table 2.5: Search Keywords for RQ2

Domain	Keywords
Email Data Features	("email metadata" OR "email content" OR "email features" OR "email attachments")
Phishing Detection	("phishing detection" OR "phishing datasets" OR "phishing" OR "phishing attacks")

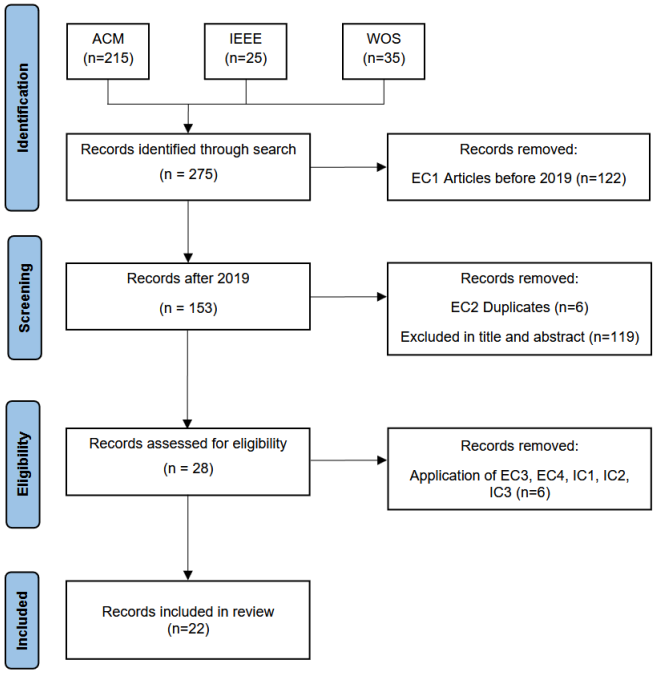

Figure 2.2: PRISMA Search Process for RQ2

works. While standard exclusion criteria (EC1-EC3) ensured the quality and relevance of the literature, the exclusion of secondary sources (EC4) was the biggest change for this review. This step was essential to ensure the findings were drawn exclusively from primary, empirical evidence. The specific inclusion and exclusion criteria are detailed in Table 2.6.

Table 2.6: Inclusion and Exclusion Criteria for RQ2

Inclusion Criteria	Exclusion Criteria
IC1: Proposes, analyzes, or validates specific email features for phishing detection	EC1: Articles published before 2019
	EC2: Duplicate articles from different databases
IC2: Presents an empirical study or a system implementation	EC3: Articles not published in English
IC3: Provides a clear contribution to the state of the art	EC4: Secondary literature (surveys, reviews) and grey literature (theses, book chapters)

2.2.2 Findings and Discussion

A total of 275 records were initially retrieved for RQ2 by applying the search query to the selected databases. After filtering out publications from before 2019, 153 records remained. During the screening phase, 6 duplicate records were removed, and 119 articles, primarily proceedings, were excluded based on their titles and abstracts. This left 28 records for further eligibility assessment. At this stage the exclusion criteria (EC3, EC4) were applied, resulting in the exclusion of 6 additional records. Ultimately, 22 publications were selected, consolidated, and systematised in the review. The PRISMA flowchart summarising this process is presented in Figure 2.2.

The analysis of phishing emails has identified several critical and replicable features that distinguish them from legitimate correspondence [99, 100]. These features can be categorised into distinct groups, including the linguistic content of the email body, the structure and presentation of embedded Uniform Resource Locators (URLs), sender information and header metadata, and the use of psychological manipulation tactics.

The content of the email body itself is the main source of evidence for phishing detection. Advanced Natural Language Processing (NLP) models have proven highly effective at identifying malicious patterns in text. For instance, using BERT vectors to analyse email subjects and bodies, has been shown to significantly improve the clustering of phishing campaigns based on shared attributes like the type of fraud or the entity being impersonated [101]. The power of text-based features can also be generalisable across different languages. Research by Salloum et al. [102] showed that a model using Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction and an Multi Layer Perceptron (MLP) classifier could achieve high accuracy on a bilingual English-Arabic corpus, confirming that linguistic patterns are a robust indicator regardless of language. Other approaches using Word2Vec embeddings with ML models have also yielded strong results, highlighting that the email body contains sufficient information for detection without needing to analyse personal data, which helps to address privacy concerns [103]. Furthermore, frameworks combining BERT for feature extraction with a Convolutional Neural Network (CNN) for classification have reached high accuracy by focusing exclusively on linguistic features, showing the capability of Transformer-based models in this domain [104].

Embedded URLs and links are another highly significant indicator of phishing. Research into user behaviour has revealed that the presentation of these links heavily influences susceptibility. Attackers often exploit the fact that users may not inspect a URL until after they have already clicked it, a vulnerability that is amplified when links are masked as buttons or hypertext rather than being displayed as raw text [105, 106]. The display of these features is also impacted by the device used, as mobile platforms often obscure sender details and full URLs, thereby increasing user vulnerability [107]. Beyond visual presentation, the structural features of the URL itself are critical. For example, a comparative study by [108] found that ML algorithms such as JRip and j48 could achieve over 94% accuracy in phishing URL detection by analysing a set of 87 distinct lexical and host-based features. This focus on deep feature analysis for URLs is further supported by numerous studies showing that various Deep Neural Networks (DNNs), including CNNs and LSTMs, can effectively classify phishing URLs with high accuracy on benchmark datasets like ISCX-URL-2016 [109] and Ebbu2017 dataset [110] [111].

The email's header, particularly the sender's address and other metadata, provides a set of features for detection. Studies have shown that users who carefully inspect the sender's email address are significantly less likely to fall victim to an attack, as the domain and username can reveal signs of illegitimacy [106]. Automated frameworks have automated this by evaluating sender-specific attributes, such as the use of "no-reply" usernames or whether the sender's domain is on a trusted list [112]. More advanced DL models have demonstrated state-of-the-art performance by applying architectures like CNNs and BERT to analyse the full email header, achieving an AUC of 0.993 and proving the value of a segment-specific analysis [99].

Finally, phishing attacks are also characterised by their usage of psychological manipulation and a specific set of content-based features. Research has identified persuasive cues such as manufactured urgency, appeals to authority, and claims of scarcity as common tactics used

to manipulate users [113]. Other cognitive manipulation tactics include offering monetary incentives or using emotional triggers [112]. The most robust detection systems often combine these different feature sets into an hybrid approach. For example, a model which integrated content features, like the presence of HTML forms with textual features, achieved 99.87% accuracy [114]. Similarly, other successful approaches have combined header and body features using LSTMs [115] or have integrated sentiment analysis with LLMs to achieve high detection rates [116]. These combined approaches show that a comprehensive analysis, incorporating technical, linguistic, and psychological features, can be a good strategy for characterising and detecting phishing emails.

2.2.3 Summary of Critical Email Features for Phishing Detection

The preceding analysis identified several distinct categories of features critical for characterising and detecting phishing emails. Table 2.7 summarises these key feature sets, detailing the specific indicators discussed in the literature and the primary challenges associated with their use in detection systems.

Table 2.7: Summary of Critical Features in Emails for Phishing Detection

Feature Category	**Key Indicators Discussed in Literature**	**Primary Challenges and Applications**
Header and Metadata	Sender's address (domain, username patterns), "no-reply" strings, and analysis of the full email header metadata.	Application in detecting sender spoofing and impersonation. Challenged by email protocol limitations and how different devices display header information.
URLs and Links	Visual presentation (masked vs. raw text), lexical and host-based features of the URL string, and domain reputation.	Crucial for identifying malicious destinations. Attackers use obfuscation and redirection techniques, and user susceptibility is high, especially on mobile devices.
Email Body Content	Linguistic patterns, stylistic cues, impersonation themes, and structural elements like HTML forms.	Enables deep content analysis to find malicious intent. Prone to evasion through sophisticated language, but highly effective for robust detection.
Psychological Cues	Use of urgency, authority, scarcity, monetary incentives, and other emotional triggers.	Application in detecting social engineering tactics. Requires models that can understand nuanced, persuasive language beyond simple keywords.

The exploration of email features for phishing detection reveals a versatile approach. While technical indicators from the header and embedded URLs remain vital, state-of-the-art systems increasingly rely on analysing the email body for linguistic patterns and psychological cues to enhance detection accuracy. However, attackers continue to evolve their methods, employing techniques such as sophisticated URL obfuscation and highly convincing impersonation tactics, which challenge existing detection models.

The integration of these features into hybrid models shows the potential for building resilient systems. Despite these advancements, future efforts must address the challenges of

computational efficiency and the need for deeper contextual understanding to improve the real-world applicability and scalability of these detection frameworks.

2.3 Evaluating the Quality and Performance of Synthetic Datasets

This section addresses RQ3: "*How can the quality of synthetic data and their impact on performance of phishing detection models be effectively evaluated?*". To answer this, the research explores the established metrics and methodologies used to assess two distinct areas, the quality of the generated data and the resulting performance improvements in phishing detection models.

2.3.1 Methodology

The investigation for RQ3 followed the same systematic review approach guided by the PRISMA framework as RQ1 and RQ2. This process began with an initial screening of titles and abstracts, followed by a detailed full-text review of articles to determine their final eligibility. Only publications that met the criteria were included in the final review.

The search strategy was designed to identify research studies that combined three domains: the usage of synthetic data, phishing detection, and the methods to evalute performance. To achieve this, keywords for each domain were combined. Terms like "synthetic data" and "generated data" were used to capture the main subject. These were paired with terms such as "phishing detection". Finally, keywords like "performance" and "metrics" were included to ensure the selected studies discussed evaluation. The term "email" was added as a filter to narrow the scope of the communication channel relevant to this thesis. Table 2.8 provides a summary of the complete search terms.

Table 2.8: Search Keywords for RQ3

Domain	Keywords
Synthetic Datasets	("synthetic data" OR "synthetic dataset" OR "artificial data" OR "generated data")
Phishing Detection	("phishing detection" OR "phishing dataset" OR "phishing")
Metrics	("performance" OR "metrics" OR "evaluation")
Email	("email")

The same databases used for the previous research questions, IEEE, ACM and WoS, were employed for the search addressing RQ3. Recognising the specialized nature of this topic, the initial search yielded a limited number of studies. To ensure a better review, the search was expanded to include the Springer database [117], which is also a prominent repository for computer science and AI research. The search query focused on publications from 2019 onward and were restricted to those available in English.

To ensure the quality and relevance of the final selection, inclusion and exclusion criteria were used. To be included, a study had to meet two conditions, it needed to discuss performance metrics for phishing detection (IC1) and make use of synthetic data (IC2). Alongside standard criteria for relevance, duplication, and language (EC1-EC3), secondary

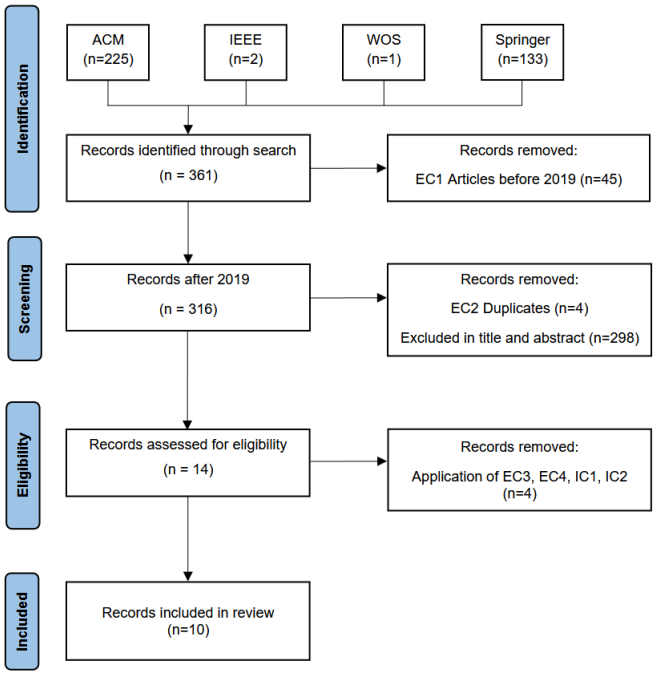

Figure 2.3: PRISMA Search Process for RQ3

literature like surveys and reviews were also excluded (EC4). The full criteria are detailed in Table 2.9.

Table 2.9: Inclusion and Exclusion Criteria for RQ3

Inclusion Criteria	Exclusion Criteria
IC1: Performance metrics or evaluation methods for phishing detection models IC2: Usage of synthetic data	EC1: Articles published before 2019 EC2: Duplicate articles EC3: Not being in English EC4: Secondary literature (surveys, reviews) and grey literature (theses, book chapters)

2.3.2 Findings and Discussion

A total of 361 records were initially retrieved for RQ3 by applying the search query across the selected databases. After filtering out publications published before 2019, 316 records remained. During the screening phase, 4 duplicate records were removed, and 298 articles were excluded based on their titles and abstracts. This left 14 records for further eligibility assessment. At this stage the exclusion criteria (EC3, EC4) were applied, leading to the exclusion of 4 additional records. Ultimately, 10 publications were selected, consolidated, and included in the review. The PRISMA flowchart summarizing this process is presented in Figure 2.3.

The quality of synthetic data for phishing detection is evaluated not on its own, but by measuring its impact on the performance of detection models. An effective evaluation, therefore, involves generating synthetic data to address a specific problem, such as class

imbalance or adversarial robustness, and then using a comprehensive suite of metrics to quantify the resulting improvements [10, 118, 119].

The most common method for evaluating the impact of synthetic data is to use it to balance a dataset and then measure the performance gains on standard classification metrics. Accuracy, precision, recall, and F1-score are consistently used as the primary indicators of a model's effectiveness [111, 120, 121]. A study by [118] provides a clear example of this process. The authors first generated adversarial phishing emails using tools like TextFooler to create subtle word- and character-level perturbations, and then fine-tuned a GPT-2 model to produce entirely new synthetic emails. The quality of this generated data was then validated by training an ALBERT-based detector on datasets with progressively balanced phishing-to-legitimate ratios. As shown in Table 2.10, the results indicated a clear positive correlation: as the proportion of synthetic data increased towards a 1:1 ratio, the F1-score increase from 0.94 to 0.97. The inclusion of the Matthew's Correlation Coefficient (MCC) in their evaluation is also significant, as its rise from 0.93 to 0.97 further confirms that the synthetic data was of high quality and genuinely improved the model's discriminative power.

Table 2.10: Performance Metrics at Different Data Ratios

Data Ratio	Accuracy (Acc)	F1 Score	Precision	Recall	MCC
1/8	0.988	0.94	0.98	0.91	0.93
1/4	0.990	0.95	0.94	0.96	0.95
1/2	0.991	0.95	0.95	0.96	0.95
1/1	0.994	0.97	0.98	0.96	0.97

This evaluation strategy, which validates synthetic data through the performance gains it enables, is reinforced by other research. For instance, the work by [10] employed a three-phase approach of generating data with the LeakGAN model, assigning probabilistic labels to the newly created data using a Positive-Unlabeled (PU) learning method, and finally classifying it with BERT. They explicitly chose the F1-score as their primary metric due to its robustness in imbalanced scenarios. Similarly, [119] demonstrated the quality of their GAN-generated data on benchmark datasets like ISCX-URL-2016 [109] and PhishTank by showing that a classifier trained on it outperformed other deep learning methods, partly thanks to using a White Shark Optimization algorithm for intelligent feature selection.

Beyond improving performance on imbalanced data, a more advanced evaluation of synthetic data quality involves assessing its ability to enhance a model's robustness and generalisability. One approach involves generating subtle adversarial perturbations and then measuring if training on this augmented data makes the model more resilient to future, unseen attacks [118]. Another effective evaluation method is to compare how different model architectures respond to the synthetic data. For instance, a study by [122] used GPT-4 to generate synthetic emails and URLs and then evaluated different models as detectors. They found that while the DeBERTa V3 model performed best on original data, LLMs like GPT-4 and Gemini 1.5 achieved superior precision and recall on the augmented datasets, suggesting that the quality of synthetic data can be model-dependent. The quality can also be evaluated at a feature level with one study using the F1-score to assess how well synthetic data could replicate the subtle linguistic fingerprints of genuine deception as captured by the LIWC feature set [123].

Ultimately, a comprehensive evaluation framework relies on a combination of these methods. It typically involves using a standard suite of metrics, with a particular emphasis on precision and recall to minimise false positives and negatives, which are critical in security contexts [124]. The effectiveness of the synthetic data is validated by demonstrating significant performance improvements on established datasets, proving that the generated data has successfully taught the model to better distinguish between malicious and legitimate content while also being realistic enough to ensure the model learns generalisable features.

2.3.3 Summary of Evaluation Metrics for Synthetic Data in Phishing Detection

To systematically measure the impact of synthetic data as discussed previously, a specific set of evaluation metrics is required. Table 2.11 provides an overview of the primary metrics used to assess the performance of phishing detection models trained on augmented datasets. These metrics are crucial for quantifying improvements in detection accuracy, robustness, and the ability to handle the severe class imbalance inherent in real-world email data.

Table 2.11: Key Evaluation Metrics for Phishing Detection Models

Metric	**Definition and Usage**	**Application and Challenges in Phishing Detection**
Accuracy	The ratio of correctly classified emails (both phishing and legitimate) to the total number of emails.	A simple starting point, but can be highly misleading in imbalanced datasets
F1-Score	The harmonic mean of precision and recall, providing a single score that balances both concerns.	Robust to class imbalance and evaluates a model's ability to both find threats and avoid false alarms.
Precision	The ratio of correctly identified phishing emails to all emails classified as phishing.	High precision means that when an email is flagged as a phish, it is very likely to be one (low false positive rate).
Recall (Sensitivity)	The ratio of correctly identified phishing emails to the total number of actual phishing emails.	High recall ensures that a high proportion of actual phishing threats are successfully caught (low false negative rate).
MCC	A correlation coefficient between the observed and predicted classifications, accounting for all four values in the confusion matrix.	Considered a very reliable and balanced metric, especially for severely imbalanced datasets, as it is not biased towards the majority class.
AUC	Measures a model's ability to distinguish between classes by plotting the true positive rate against the false positive rate at various thresholds.	Provides a threshold-independent view of model performance.

Evaluating the utility of synthetic data requires a dual-focused approach. The metrics in Table 2.11 provide an extrinsic evaluation, measuring the downstream impact on the detection

model's performance. However, a complete assessment should also include an intrinsic evaluation of the generated text itself, using metrics designed to assess text quality, coherence, and diversity. A successful synthetic dataset must not only improve classifier performance but also be indistinguishable from real-world phishing attempts to ensure the model is learning relevant, generalisable features.

2.4 Chapter Remarks

In this chapter, the state of the art in text generation, augmentation, and phishing detection was systematically reviewed, aiming to address three research questions. For RQ1, the discussion followed the evolution of text generation methods from classical techniques to modern DL approaches. For RQ2, the review identified the key characteristics that distinguish phishing emails. Finally, for RQ3, the analysis examined how synthetic datasets are evaluated in the phishing detection domain.

The review of text generation and augmentation techniques revealed a clear progression. Early approaches, though easy to interpret, were limited when applied to larger or more complex tasks. Models such as RNNs and LSTMs were among the DLs models that could generate text with some fluency, but they struggled with longsentences. The introduction of the Transformer addressed many of these weaknesses and turning possible the PLMs and LLMs, which are now widely used for creating coherent text. Alongside these methods, other paradigms have been investigated, including GANs, VAEs, and diffusion models, each bringing different strengths as well as technical challenges, such as training instability which limited their widespread adoption for this type of task.

The focus of modern text generation has expanded beyond mere fluency to include factual accuracy and contextual relevance. This evolution in generative techniques has also driven a shift in evaluation methods. While metrics like BLEU and ROUGE are still prevalent for measuring lexical similarity, they are increasingly supplemented by semantically aware metrics like BERTScore, which better capture a text perceived quality.

Attackers continue to evolve their techniques, making effective phishing detection a challenge that relies on analysing a combination of indicators. These features span the entire email, from header data and embedded URLs to the message content and the psychological tactics employed. Although modern architectures like BERT and CNNs can process these diverse inputs, the literature confirms their ongoing difficulty in keeping pace with the new techniques used by attackers.

For RQ3, it was discussed that the evaluation of synthetic data is done by its quantifying the effects on model performance. Metrics such as the F1-score and MCC, which handle class imbalance reliably, are often used for this purpose. These differ from the metrics applied in RQ1, where the focus was on the intrinsic quality of generated text through metrics such as ROUGE and BERTScore. As such, the findings suggest that a full evaluation should consider both the quality of the text itself and the impact it has on the detection of a model.

Approaches based on GANs and adversarial augmentation provide a way to address data scarcity, but generating realistic and domain specific phishing content remains challenging. The current methods still fall short of producing emails that capture the structural patterns of real emails while also appearing psychologically convincing and contextually appropriate. Closing this gap will require specialised techniques designed to create high-fidelity synthetic

data that goes beyond grammatical correctness to mirror the tactics used in real-world phishing attempts.

Chapter 3

Data Protection and Ethics

3.1 Ethical and Regulatory Framework

This research, centered on the generation of synthetic email data, operates at the intersection of data privacy and AI ethics. Emails are prone to have sensitive data, often containing Personal Identifiable Information (PII) that can be linked to individuals. Consequently, a principle of this study is the commitment to ethical and responsible innovation, which works on top of a robust regulatory framework. Our approach is guided by two pieces of European legislation: the General Data Protection Regulation (GDPR) [125] for data handling and the AI Act [126] for the governance of the system itself.

3.1.1 Data Protection under the GDPR

The primary responsibility is to safeguard the privacy of individuals whose data is used in this research. To this end, we strictly adhere to the principles of the GDPR [125]. When working with public, open-source datasets where direct user consent is not feasible, the ethical duties are taken into consideration by implementing data protection measures.

To mitigate privacy risks, we employ pseudoanonymisation and data minimisation techniques. Rather than using real personal details, we replace sensitive information, such as names and other PII, with plausible but entirely fictitious dummy text. This method allows us to preserve the linguistic coherence and contextual integrity of the data, which is essential for training our models, without compromising individual privacy.

Additionally, to prevent any misuse of the data, access to both the original datasets and the synthetic data generated by our system is strictly controlled. In compliance with Article 89 of the GDPR, which provides safeguards related to processing data for scientific research purposes, data is made available only to authorised researchers directly involved in this project. All methodologies, data handling procedures, and decisions are comprehensively documented to ensure full transparency and accountability.

3.1.2 Compliance with the AI Act

Building on this foundation of data protection, our framework extends to the responsible development of the AI system itself. We proactively address the guidelines set forth in the European Union (EU)'s landmark AI Act [126], which establishes a proportionate, risk-based approach to regulating AI.

A critical first step in our analysis is to classify the system according to the Act's risk pyramid. Our contextually-aware email generator clearly falls under the definition of an 'AI

system' as per Article 3(1), as it is a machine-based system that generates content capable of influencing virtual environments. However, its intended purpose does not align with any of the prohibited practices listed in Article 5, such as social scoring or subliminal manipulation, meaning it does not pose an 'unacceptable risk'.

Furthermore, an evaluation against the high-risk categories enumerated in Annex III of the AI Act shows that our system is not 'high-risk' by default. These use cases are highly specific, covering areas like critical infrastructure, employment, and law enforcement. Our system is designed as a general tool for synthetic data generation for research, not for a specific high-stakes application.

Therefore, our synthetic email generation system can be classified as a limited-risk AI system. This classification exempts it from the extensive conformity assessments required for high-risk systems but imposes crucial transparency obligations under Article 50. These obligations are important for building trust and mitigating the potential for misuse, such as disinformation.

As mandated by Article 50(2), providers of AI systems that generate synthetic text must ensure the outputs "are marked in a machine-readable format and detectable as artificially generated or manipulated". To fulfil this, any system resulting from this research should embed a machine-readable marker in its output, which allows for automated detection without altering the human-readable body of the message.

While this research is conducted in a controlled environment, the principle of transparency remains central. Should any technology developed from this work be deployed in a context where it interacts with people, it is required to disclose its artificial nature to any affected persons, aligning with the objective of Article 50(1).

In summary, this study is conducted with a comprehensive approach to regulatory compliance. By integrating the data-handling protocols of the GDPR with the transparency requirements of the AI Act, we ensure that our work is not only technologically innovative but also ethically sound and aligned with European values. This dual commitment strengthens the credibility of the research and presents our dedication to develop trustworthy AI.

3.2 Usage of Generative AI

To guarantee full transparency and academic integrity, this section discloses the role of Generative AI tools in the preparation of this thesis. Generative AI was used strictly as an assistive tool to support, my own research and writing process. I as author retain full responsibility for the intellectual integrity, critical analysis, and final content of this work.

During the writing process, my original, unstructured, ideas were provided to a Generative AI model to help organise the information into a more coherent narrative flow. This process was also used to enhance the text by improving grammatical correctness, suggesting alternative phrasing for clarity, and ensuring a consistent academic tone throughout the document.

Technical assistance during the development of the framework, was also done with the help of Generative AI. AI was employed to help debug code snippets, particularly for issues where online searches and documentation weren't enough. This served to accelerate the development process and deepen the my technical understanding of the underlying software.

I established a clear boundary for the AI's role, using it as an assistant rather than a primary researcher. The core intellectual work of this thesis, from formulating the research questions

and conducting the literature review to analysing the data and results, and drawing the final conclusions, is entirely my own.

Chapter 4

Proposed Framework

To enhance the robustness and generalisation capabilities of ML models in phishing detection, this chapter introduces **Context-Aware Narrative Development for Adversarial Communication Emails (CANDACE)**, a novel framework for generating synthetic emails. The proposed framework is designed to significantly augment the diversity of training data by creating realistic and contextually relevant email messages. This chapter details the framework's overall architecture, explains the function of each of its core components, and discusses how it operates within a larger system, following the C4+1 model [20], and the integration of these components, that had inspiration on the CRISP-DM [21] methodology.

4.1 Architectural Overview

This section presents the architecture of the proposed framework for generating context-aware emails. The primary goal of this architecture is to generate syntethic and context-aware email messages, to enhance the robustness and generalisation of ML models used for phishing detection.

The CANDACE framework works as the adversarial component within the bigger PERRY system, whose architecture is shown in Figure 4.1. PERRY itself is a multi-stage phishing detection pipeline designed to analyse incoming emails, by extracting features from their headers, content, and URLs, and uses ML and DL models and LLMs to classify them as malicious or benign. While the PERRY system is designed to detect these threats, CANDACE role is to create these threats. It generates synthetic, context-aware emails that are used for training and testing the system phishing detection models. As highlighted in the diagram, it is the engine that provides the challenging data needed to make the detection components smarter and more resilient. This synthetic data is used in conjunction with real-world emails to comprehensively train and validate the detection models.

The internal architecture of CANDACE is designed as a modular pipeline, as illustrated in Figure 4.2. It consists of three main components: the Email Generator which is the core of it, the Knowledge Graph (KG) and the context.

The generation process starts by receiving the initial context. The Email Aggregator then manages the flow of data between modules, beginning by forwarding the context to the Context Knowledge Graph component, the framework's contextual "brain". The process begins with the Context Knowledge Graph module, which is responsible for transforming a high-level theme into a factually grounded scenario. This initial module produces a structured context block containing a designated sender, recipient, and an enhanced context. This structured information then serves as the primary input for the subsequent generative

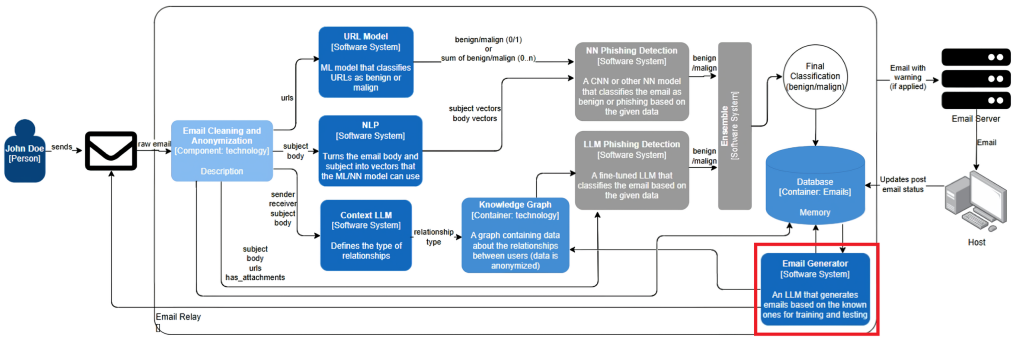

Figure 4.1: Architecture of the PERRY system. The red box highlights the CANDACE framework developed as the core contribution of this thesis.

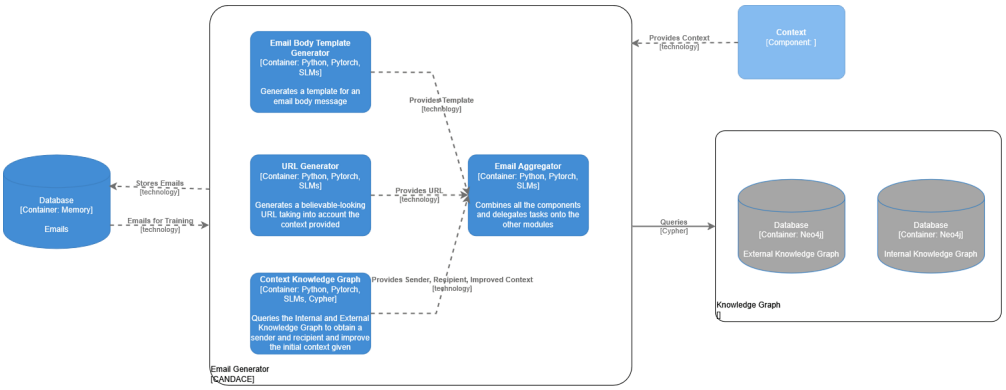

Figure 4.2: Architecture of the CANDACE Framework

components. The Email Body Template Generator uses the context to produce a coherent body text template with placeholders for dynamic information. Using the same context as the module before, the URL Generator is tasked with creating believable-looking links. Finally, all of these elements join together at the Email Aggregator. This component acts as the final assembler, integrating the template from the Body Generator, using the improved context from the Context Knowledge Graph, and using the link from the URL Generator to create a contextual, coherent and fluent email message.

4.2 Email Body Template Generator

The generation of contextually aware and linguistically coherent email messages is the main objective of the CANDACE framework. Given the inherent complexity of maintaining semantic coherence while integrating psychological cues and contextual tags, this module employs PLMs due to their advanced natural language generation capabilities, as it was noted in Section 2.1.2. Alternative generative approaches, including GANs, were also evaluated but, they proved less effective for generating long text sequences due to their inherent difficulty in maintaining semantic coherence. As a result, PLMs emerged as the preferred solution given their proficiency in processing large corpora, retaining contextual information through attention mechanisms, and with their capability to incorporate external knowledge from structured sources, such as KG [32], to make them contextually aware. At its core, a KG is a structured representation of knowledge that models a domain as a network of entities

and the relationships between them. Information is stored in the form of "triples", with each triple consisting of a `subject-predicate-object` structure, such as `John Doe - works at - Microsoft`.

Unlike a traditional database that stores isolated data points, a KG is designed to capture the context of how entities are interconnected. For example, a list of employees, is modelled as an entire web of connections, which employee works in which department, what that department's function is, and who manages it. By organising information in this way, the KG transforms a collection of facts into a cohesive, more readable manner of a real-world environment.

PLMs encompass architectures of varying scales, including Small Language Models (SLMs) and LLMs, differentiated primarily by their parameter count, computational requirements, and accessibility constraints. Model selection followed two fundamental criteria: first, models needed to be open-source or freely available to ensure accessibility and reproducibility; second, for comparative analysis the models needed to have a similar parameter count to evaluate performance objectively. Based on these criteria, GPT-2 (1.5B) from OpenAI [127] and Gemma 3 (1B) from Google [128] were selected.

The two models were chosen to provide a comparison between a legacy architecture and a modern counterpart. GPT-2 was selected as a baseline, with its pioneering role as an early large-scale Transformer and its well-documented performance make it an ideal point of comparison. In contrast, Gemma 3 represents the current state-of-the-art in large-scale model design. As a distilled version of Google's larger Gemini model, it incorporates the latest training methodologies and architectures.

This section details the complete methodology for the Email Body Template Generator module. It will first cover the extensive data curation process required to train models for generating both deceptive phishing and realistic benign email content. Following this, it will present the comparative workflow and specific fine-tuning strategies applied to the selected SLMs to achieve this dual objective.

4.2.1 Data Preparation

To develop, train and evaluate the module, a diverse collection of open-source datasets was compiled and carefully preprocessed. This corpus includes both phishing and benign emails, enabling the models to learn a wide range of linguistic patterns and contextual cues. The primary sources include the Enron Corpus [129], the Nazario Phishing Archive [130], the Nigerian Fraud Corpus [131], datasets from the Text REtrieval Conference (TREC) [132–134], and the CEAS 2008 collection [135].

The Enron Corpus is a repository of emails from the now-defunct Enron Corporation. It contains approximately 517,401 messages from about 150 users, primarily senior management, organised into folders. These emails span from 2000 to 2002, providing a view of corporate communications. The dataset was originally made public by the Federal Energy Regulatory Commission during its investigation into the company's practices. The Nazario Phishing Archive curated by security researcher Jose Nazario, comprises phishing emails collected over nearly two decades, from 2005 to 2024. The dataset offers a broad spectrum of phishing tactics and schemes employed over the years, containing around 9,110 email messages. The Nigerian Fraud corpus focuses on advance fee fraud emails, commonly known as "Nigerian prince" scams. These emails typically promise large sums of money in exchange for an upfront fee. The dataset includes examples from the early 2000s, consisting of 3,975 email.

The Text REtrieval Conference (TREC) has organised several tracks focusing on spam detection, notably in 2005, 2006, and 2007. The 2005 TREC Public Spam Corpus, for instance, contains 92,189 email messages, with 52,790 labeled as spam and 39,399 as ham. The Conference on Email and Anti-Spam (CEAS) in 2008 introduced datasets that include modern examples of spam and social engineering attacks. These emails reflect the tactics employed by spammers and phishers around that time it consisted on around 39,154 email messages. Figure 4.3 illustrates the data distribution of the previously described datasets.

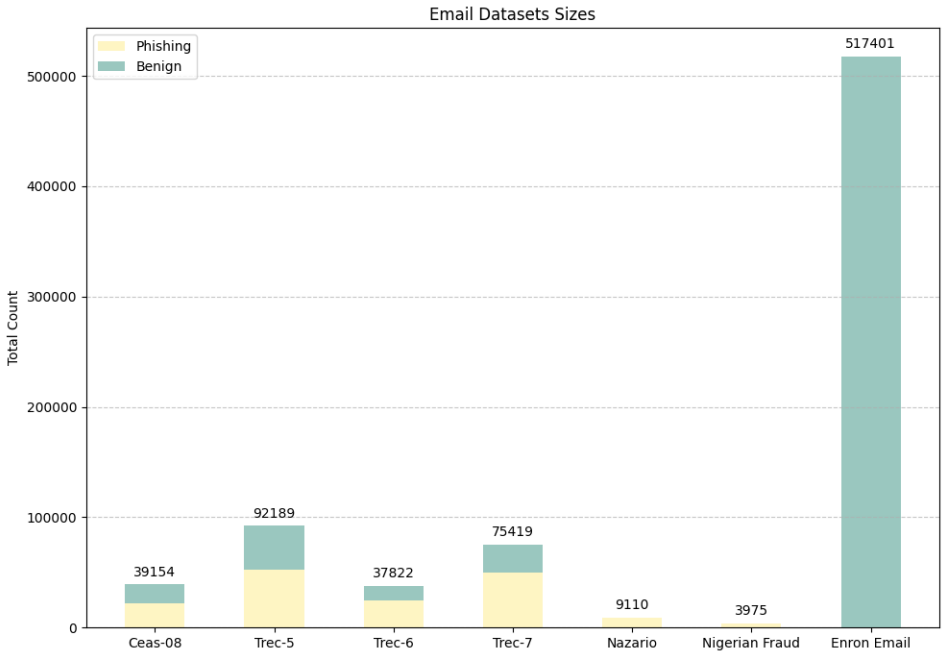

Figure 4.3: Distribution of Phishing and Benign Email Messages across Datasets

Table 4.1 compares several datasets to find suitable ones for this project. The goal was to find data that is realistic for training. We checked for authentic domain representation, content structure, and full email elements, because these details can show whether a dataset mirrors real-world emails. We also assessed the existence of complete labels and verified phishing samples to ensure the data was reliable enough to train a model.

Table 4.1: Comparison of Datasets for Phishing Detection

Dataset	**Realistic Domains**	**Realistic Content**	**Labeled Dataset**	**Complete Capture**	**Contains Phishing**
Enron	Yes	Yes	No	Yes	No
Nazario	Yes	Yes	Yes	Yes	Yes
Nigerian Fraud	Yes	Yes	Yes	Yes	Yes
TREC	Yes	Yes	Yes	No	Yes
CEAS	Yes	Yes	Yes	No	Yes

The age of the datasets used, namely the Enron Corpus and the Nigerian Fraud collection, were taken into consideration on this research. These legacy datasets may not fully reflect the linguistic patterns and stylistic conventions of modern emails. This temporal gap can, however, be effectively bridged through the use of modern PLMs. Since these models

were pre-trained on vast and recent text, they possess an inherent understanding of contemporary language. This knowledge transfer gives the model the ability to apply modern linguistic insights to older data, thereby mitigating the limitations caused by the scarcity of contemporary email datasets.

To ensure ethical compliance and prepare the data for effective model fine-tuning, the raw emails were processed through a six-stage pipeline designed to transform them into generalised email templates, as shown in Figure 4.4. These templates act as skeleton structures, where specific details are replaced by placeholders, allowing a model to learn the structural and linguistic patterns of malicious and benign correspondence, rather than memorising specific entities.

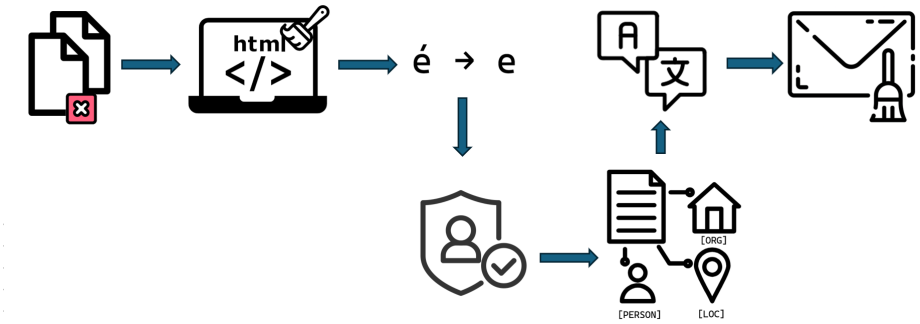

Figure 4.4: Six Stage Preprocessing Email Template Pipeline

The pipeline began with the removal of duplicates, both within and between the datasets, to establish a diverse starting point. Second, structural sanitisation was applied to remove HTML/CSS/ASCII artifacts, followed by Unicode normalisation to remove ambiguities like homoglyphs. Next, direct identifiers such as email addresses, URLs, and phone numbers were masked using regular expressions, replacing them with standardized tags (`[EMAIL]`, `[URL]`, `[TELEPHONE]`).

The fourth stage involved pseudoanonymisation using Named Entity Recognition (NER). This step was designed to remove contextual and personal identifiers that regular expressions would miss, and weren't able to capture. Using SpaCy's medium-sized English model, entities such as personal names, organizations, and geopolitical locations were identified and replaced with placeholders (e.g., `[PERSON]`, `[ORG]`, `[GPE]`). This process is termed "pseudoanonymization" because while it removes explicit named entities, it does not guarantee complete anonymity, as other contextual information within the email body could potentially be identifying. Its primary purpose here is to abstract the content, to force the model to generalise beyond names and entities.

The final stages included language filtering with FastText to retain only English emails and a concluding deduplication step. This resulted in final datasets of 49,146 unique phishing templates and 56,644 unique benign templates. The significant reduction in size is primarily due to the removal of the entire Enron-derived email set. This dataset was removed due to a substantial portion of this corpus was present in both the phishing and benign source collections, creating a data contamination issue. Manually classification of these emails was deemed time-consuming, making complete removal the only method to ensure data integrity.

As it was mentioned, the output of this six-stage pipeline is a collection of sanitized email templates. Figure 4.5 provides a concrete example of a phishing template after processing.

Notably, sensitive identifiers and specific details have been replaced with placeholders like `[URL]` and `[CARDINAL]`, while the core structure and persuasive language remain intact.

[START] Find your medication instantly! [NEWLINE]
A whole range of tablets! Take a look! [NEWLINE]
All medications in [CARDINAL] place! [NEWLINE]
[URL] Stop receiving promotional material now [URL] [END]

Figure 4.5: Example of a Sanitized Phishing Template

With the sanitized data prepared, the next step is to structure it for the model's fine-tuning process. This is achieved using a carefully engineered prompt template. Rather than feeding the sanitized text directly to the model, each email template is inserted into this prompt via the {email_message} token. The prompt serves as a wrapper, providing explicit instructions and context that guide the model on how to interpret and learn from each example.

As detailed in Figure 4.6, this prompt template was engineered to enforce strict ethical and formatting guidelines. It implements the fine-tuning by instructing the replacement of all real-world information with designated placeholders, reinforcing the sanitisation already performed. To enhance the realism of the simulated phishing attempts, the prompt requires the inclusion of psychological hooks (e.g., urgency, authority) and tolerates minor grammatical errors. Furthermore, it instructs the model to either mirror the structure of the provided example or generate original content based on a specified theme. Finally, the template enforces explicit prohibitions, such as forbidding the use of real personal information or markdown formatting, to maintain data integrity and output consistency.

The benign email messages prompt was very similar to the phishing prompt, with some differences to remove the phishing intent. These changes included removing references to phishing simulation, psychological manipulation tactics, and phishing-themed templates to fall back. Instead, the benign prompt instructed the model to generate ordinary email templates, while keeping the same placeholder variables and structural rules and prohibitions.

With the data cleaned and prepared for fine-tuning, the following section details the complete workflow of the Email Body Template Generator module, from data ingestion to model evaluation.

4.2.2 Architecture

The workflow for the Email Body Template Generator module is illustrated in Figure 4.7. The diagram visually represents the methodology of this work, beginning with a unified data preparation phase that feeds into two distinct, parallel fine-tuning paths: a full fine-tuning strategy for GPT-2 and a parameter-efficient strategy for Gemma 3. The outputs of both paths converge at a final evaluation phase, allowing for a direct comparative analysis of their performance.

The **data preparation phase**, which was detailed in Section 4.2.1, represents the initial stage of this architecture. Its output is a partitioned dataset of prompt-wrapped email templates, as it was mentioned, which is then split into training (80%) and test (20%) subsets.

During the **model fine-tuning phase**, both selected SLMs, GPT-2 (1.5B) and Gemma 3 (1B), underwent specialised fine-tuning. While an earlier iteration of the project considered other models like Gemini 2.0 Flash, Gemma 3 was ultimately selected as the modern

You are an AI phishing simulation specialist creating training materials for email security detection systems. Generate email bodies that follow these strict rules:
- If provided with an example email: Mirror its structure/style using synonyms & placeholders
- If given a theme (e.g., "bank scam"): Create original content matching the theme
- If neither example nor theme: Use common phishing templates
- ALWAYS replace real information with these EXACT variables: [URL], [EMAIL], [DATE], [ORG], [PERSON], [CARDINAL], [TIME], [MONEY], [PRODUCT], [LAW], [GPE], [QUANTITY], [PERCENT], [FILE], [SIMBOL]
- Use [NEWLINE] for paragraph breaks
- [PAD] ONLY after [END] for length consistency
- Psychological hooks/cues are required (urgency, consequences, reward, authority)
- Can have minor grammatical errors
- Avoid technical jargon
- Keep text coherent

You are Prohibited:
- Using real names/numbers
- Using markdown formatting
- Using [PAD] before [END]

Email Example:
{email_message}

Theme:
{start_theme}

Response:
{response}

Figure 4.6: Phishing Prompt for Email Body Template Fine-Tuning

counterpart to GPT-2. This decision was based on its comparable parameter scale and contemporary architectural features, which allows for a more direct and meaningful comparison against the legacy GPT-2 model.

Divergent fine-tuning strategies were applied to each model. GPT-2 underwent *full fine-tuning* across both phishing and benign datasets, yielding two distinct variants capable of replicating the structural and linguistic nuances specific to each email classification. Implementation employed the AdamW optimiser with a learning rate of 5×10^{-5}, batch size of 16, and mixed-precision training (FP16) on NVIDIA A4500 hardware, completing over 10 epochs.

In contrast, Gemma 3 underwent *parameter-efficient partial fine-tuning* via the Unsloth framework, a library designed to accelerate training while significantly reducing memory usage [136]. In this approach, only 1.3% of the model's parameters were tuned. This selective method preserves the model's generalisation capabilities while efficiently adapting it to the email generation task. Training utilised identical hardware and precision settings, with a batch size of 128, a learning rate of 5×10^{-4}, and was executed over 100 discrete

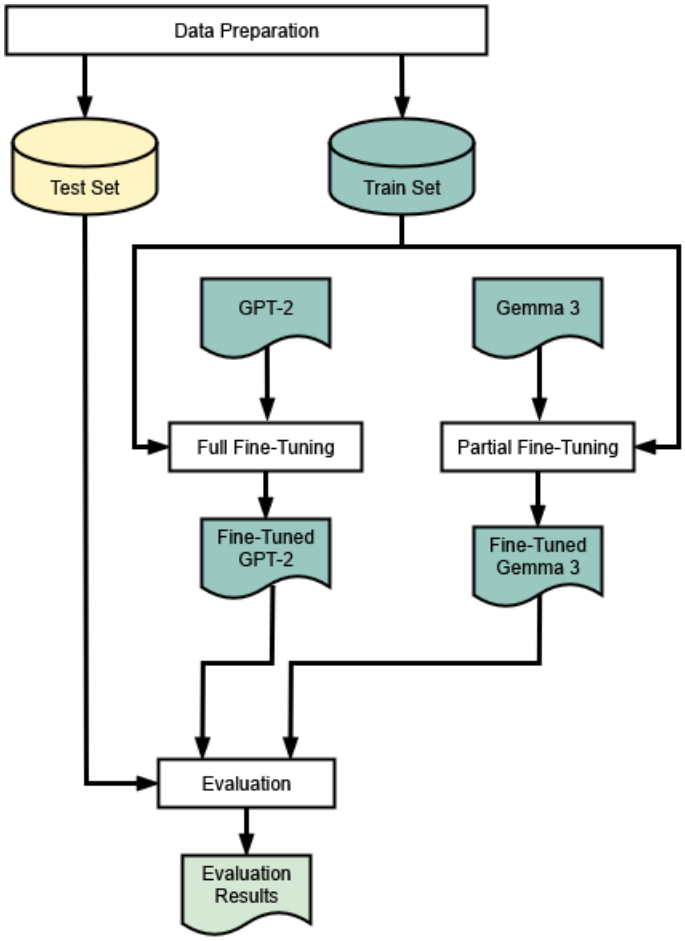

Figure 4.7: Comparative workflow for the Email Body Template Generator module. The diagram illustrates the two distinct paths: full fine-tuning applied to GPT-2, and parameter-efficient partial fine-tuning applied to Gemma 3, which converge at the final evaluation stage.

steps.

This difference in fine-tuning approaches come from technical constraints and optimisation objectives. Unsloth's partial fine-tuning was applied to Gemma 3 to minimise its memory footprint and computational duration. GPT-2 required full fine-tuning due to its incompatibility with the Unsloth framework, though its mature architecture permitted a feasible implementation.

This methodology, which combines the prompt engineering from the data preparation phase with the fine-tuning described here, creates a hybrid approach. This enforces structural compliance and ethical constraints during generation while ensuring the models can assimilate the distinct linguistic patterns of both phishing and benign examples.

The **evaluation phase** assesses the quality of the generated templates, with the detailed results presented in Section 5.1. The evaluation is conducted using a suite of established metrics, which were previously introduced and defined in Section 2.1.3. This specific set of metrics was selected to provide a assessment of generative quality, directly addressing the trade-off between realism and novelty. The first group, with ROUGE [89] and BERTScore [93], measures fidelity to the references, ensuring the generated templates are coherent and

semantically aligned with real-world examples. On the other hand, the second group, which includes Novelty [95], Diversity [95], and Distinct-N [94], quantifies the model's creative and evasive capabilities.

Finally, the refined models are deployed as core components of the broader CANDACE framework, that will be referenced further in Section 6.2. This integrated architecture not only supports the generation of synthetic emails but also contributes to the continuous improvement of phishing detection systems by providing rich, varied, and realistic training data. Upon implementation, the framework will keep incorporating new synthetic data to improve ML model robustness while assimilating genuine email messages for ongoing system refinement.

4.3 URL Generator

To enhance the framework's modularity and enable specialised handling of the critical email components, a dedicated URL Generator module was developed. This component focuses exclusively on generating URLs to complement email templates generated by the Email Body Template Generator. The integration of authentic-looking URLs significantly enhances the realism of synthetic email messages while providing critical attack vectors for phishing detection systems.

Two distinct architectural approaches were evaluated for URL generation: GANs and PLMs, with particular emphasis on SLMs due to their favourable balance of computational efficiency and generative capability. Given the specialised nature of URL syntax and the resource constraints identified during preliminary testing, Gemma 3 1B was selected for the SLM approach. This section details the development of the URL Generator module, presenting the methodology in a logical progression. It begins with the data curation process needed for all approaches. It then describes two distinct architectural implementations: first, a baseline, context-free generator, and second, an experimental module that incorporates contextual awareness to produce more realistic and targeted URLs for adversarial attacks.

4.3.1 Data Preparation

The development of the URL Generator module required a diverse training corpus. This corpus was aggregated from two primary types of sources: dedicated, publicly available URL datasets, and URLs extracted directly from email corpora. The first group consists of four URL datasets: PhiUSIIL [137], a balanced dataset containing a large collection of labeled phishing and legitimate URLs, Mendeley V3 [138], a web phishing detection dataset, Kaggle 2023 [139], a dataset providing recent examples of malicious URLs and Malicious URLs Dataset [140], a repository focused on a wide variety of malicious web addresses.

As it was mentioned, to supplement these structured datasets with more examples, the corpus was further enhanced by extracting all embedded URLs from the email message collections detailed previously in Section 4.2.1. This includes links from the Enron Corpus, the Nazario phishing dataset, TREC05-07, CEAS08, and the Nigerian Fraud email repository, creating the Combined Dataset. Table 4.2 summarises the key characteristics of the primary source datasets, including their sample counts and class balance.

Duplicate detection was conducted through two distinct ways: intra-dataset analysis (within individual datasets) and inter-dataset analysis (across multiple datasets). Intra-dataset duplicates were identified via exact matching and were removed. For inter-dataset analysis,

Table 4.2: URL Dataset Composition and Characteristics

Dataset	Number of URLs	Balanced
PhiUSIIL	235,795	No
Mendeley V3	11,430	Yes
Kaggle 2023	808,042	No
Malicious URLs	651,191	No
Combined Dataset	104,786	No

we computed pairwise duplication rates using a heatmap (Figure 4.8), where each cell (i, j) represents the percentage of URLs in dataset i that also exist in dataset j. In this heatmap it is possible to observe significant overlap between the datasets, specifically:

- Mendeley V3 is fully contained in Kaggle 2023.
- 69% of Kaggle 2023 URLs overlap with the Malicious URLs dataset.
- 24% of Mendeley V3 URLs are present in the Malicious URLs dataset.

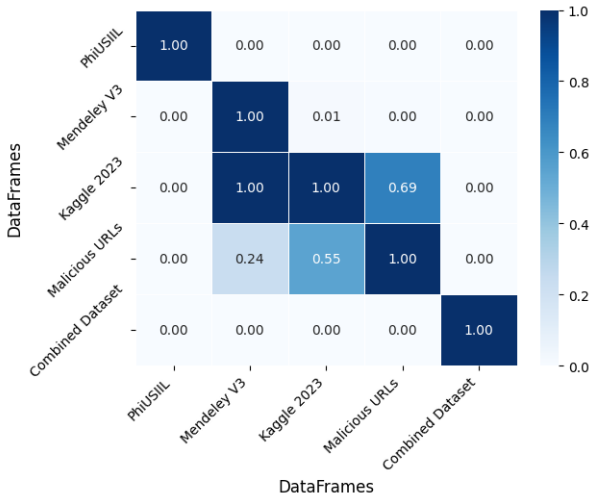

Figure 4.8: Inter-Dataset Duplicate URLs

The observed URL overlaps revealed redundancy across datasets, needing the elimination of duplicate entries. Following the removal of exact duplicates, inter and intra datasets, the refined dataset comprised 1,491,187 unique URLs. To further enhance lexical diversity and mitigate overfitting, near-duplicate detection was implemented via MinHash-based Locality-Sensitive Hashing (LSH) with a Jaccard similarity threshold of $\phi \leq 0.7$. This process reduced the dataset to 1,095,787 distinct URLs. As illustrated in Figure 4.9, the final distribution comprised of 596,687 benign and 499,200 phishing URLs, showing a slight class imbalance.

This final, dataset serves as the training and evaluation corpus for the distinct architectural approaches detailed in the following section. The benign and phishing subsets were used to train separate, specialised models within both the GAN and SLM.

4.3.2 Architecture

As it was mentioned before, the URL generation module employs two distinct architectural approaches: GANs and SLMs. Both architectures implement separate specialised models

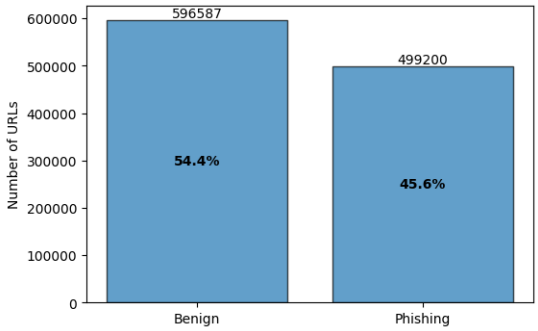

Figure 4.9: Distribution of URLs by Type after Removing Near-Duplicates from Combined Dataset

for benign and phishing URL generation to preserve domain-specific syntactic and semantic characteristics.

4.3.2.1 GAN Architecture

The GAN approach was implemented to explore its capability in generating the complex, rule-based syntax of URLs. To this end, two primary configurations were developed and compared, a baseline GAN architecture and an advanced Wasserstein GAN with Gradient Penalty (WGAN-GP), which was introduced to enhance training stability and output quality. The complete workflow for this module is illustrated in Figure 4.10.

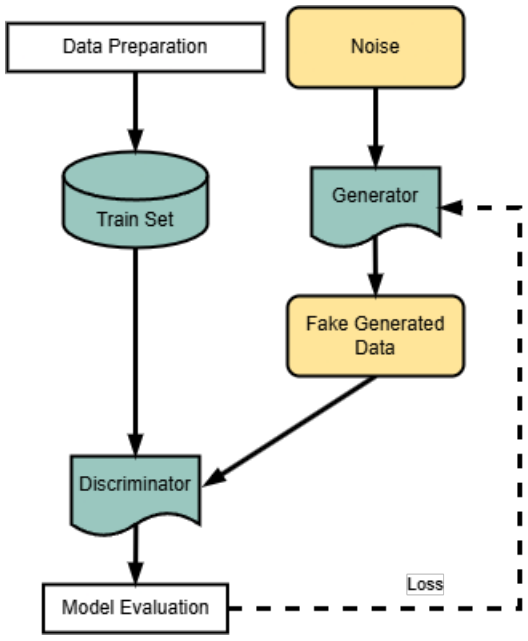

Figure 4.10: Workflow of the GAN-based URL generator. Solid arrows indicate the forward data flow through the Generator and Discriminator networks. The dashed arrow represents the adversarial loss signal, which is backpropagated from the Discriminator to train the Generator.

The process begins with a data preparation. The URL dataset is first preprocessed, with each URL string tokenized into a sequence of characters. A unified character-to-index dictionary maps each character to a unique integer, and each sequence is then padded to a fixed length

to create uniform input tensors. These tensors are then passed through a shared embedding layer that is used by both the generator and the discriminator.

The generator's role is to generate new URLs by learning the underlying distribution of the training data. Generation starts with a random noise vector, sampled from a standard Gaussian (normal) distribution, which is then fed into the generator network. This network is a MLP with three hidden layers (512, 256, and 128 neurons). To aid training, this architecture incorporates layer normalisation to stabilise activations, LeakyReLU ($\alpha = 0.2$) as the activation function, and dropout ($\rho = 0.3$) for regularisation. The final output is a sequence of logits representing the probability distribution over the character vocabulary for each position in the URL.

The discriminator's objective is to distinguish between real URLs from the training set and the synthetic ones created by the generator. To evaluate the impact of architectural choices on this task, two distinct discriminator configurations were implemented and evaluated in separate experimental runs. The first is a standard MLP with three hidden layers (128, 256, and 512 neurons), mirroring the generator's structure. The second is a CNN featuring two 1D convolutional layers (with 128 and 256 filters), designed to capture the local character patterns and sequences characteristic of valid URL syntax.

The models are trained in an adversarial loop where the discriminator's ability to classify real versus fake data is used to provide a loss signal that, in turn, trains the generator to produce more realistic outputs. To mitigate common GAN training issues like mode collapse, the WGAN-GP variant was implemented with specific modifications. This configuration reframes the discriminator as a "critic" that scores the realness of a URL and is updated more frequently than the generator (a 5:1 ratio was used). It is important to mention that a Gradient Penalty term ($\lambda = 10$) is added to the critic's loss function to enforce the Lipschitz constraint, a property that bounds the gradient of the critic's output to its input, preventing the gradients from becoming too large or erratic, which is essential for stabilising the training. Upon completion of training, the generator is used for evaluation to assess its ability to produce syntactically valid and diverse URLs.

4.3.2.2 SLM Architecture

The SLM-based approach to URL generation uses the advanced capabilities of the Gemma 3 (1B) model. The focus of this methodology is a hybrid strategy that combines parameter-efficient fine-tuning with highly structured, prompt-guided generation. This approach allows the model to learn the fundamental syntax of URLs from the training data while enforcing the specific, complex characteristics of deceptive phishing links through explicit instructions.

A component of this architecture is the engineered prompt, as detailed in Figure 4.11. This prompt serves as a set of strict instructions for the model to follow during both fine-tuning and inference. It defines a mandatory set of adversarial techniques that the generated URL must incorporate. These techniques, that are important for creating realistic phishing URLs, include the use of redirection mechanisms, IP addresses, hostname encoding, the inclusion of sensitive keywords (e.g., "login", "secure"), and the simulation of well-known brands through homoglyphs [141, 142]. By framing the task this way, the prompt ensures that the model's creative capacity is guided towards generating syntactically valid yet deceptive outputs. During the fine-tuning data preparation, each training example is programmatically inserted into this prompt structure.

You are an AI phishing simulation specialist creating training materials for email security detection systems. Generate only and ONLY a phishing URL.
MUST NOT generate links of existing companies, ONLY similar to them.
Follow these rules. Can abide by more than one rule, but is OBLIGATORY to fulfill at least one of these rules:
- Include redirection (e.g. http://3104.nnu4urye.info?http://c43n34.com?35u3b)
- The URL contains a URL of a known organization (e.g. http://108.179.216.140/~bankofamerica/)
- Special characters '-' in the host name (e.g. http://yj4yb6hmb3.x-cant-bankyou-here-of-mymoney.cn/yj4yb6hmb3/Oraliao_show_23Y)
- Long domain name (e.g. http://31837.9hzaseruijintunhfeugandeikisn.com/5/54878)
- Hostname is encoded (e.g. http://www.%64isc%72%65%74%2done-%6ei%67h%74.%63o%6d)
- IP is encoded (e.g. http://0x42.0x1D.0x25.0xC2/)
- Email address in URL (e.g http://username@hotmail.com.fddcol.com)
- IP address (e.g. http://62.141.45.54/portaleTitolaris8/)
- Suspicious symbols (@, -, ~)
- Https/ Http
- Various dots (.) in domain name
- Sensitive words (e.g. 'secure', 'account', 'webscr','login', 'ebayisapi', 'signin', 'banking', 'confirm')
- Multiple top-level-domains (e.g. http://www.ebay.com.urgd.com/path)
- Similar target brands (e.g. 'paypail', 'ebayy', etc)
- Usage of homoglyphs (e.g. cyrillic, etc)

Figure 4.11: Prompt for Phishing URL Fine-Tuning

To adapt the Gemma 3 model to the specialised task of URL generation, a parameter-efficient fine-tuning (PEFT) strategy was employed using the Unsloth framework [136]. Instead of updating all of the model's 1B parameters (full fine-tuning), this approach, using Low-Rank Adaptation (LoRA), updated only a small subset of them, 1.3% of the total parameters.

The model was fine-tuned for a total of 60 steps, a duration determined by observing the stabilisation of the training loss. The training was configured with a batch size of 8 and optimized using the AdamW algorithm with a learning rate of 1×10^{-4}. The entire process was executed on NVIDIA RTX A4500 hardware. This selective parameter-update strategy is advantageous as it allows the model to retain its vast, foundational linguistic knowledge from its original pre-training while efficiently adapting its behaviour to the specific constraints of URL syntax and the adversarial rules defined in the prompt. The overall workflow for this architecture is presented in Figure 4.12.

4.3.3 Context-Aware URL Generation

While the previous module successfully generated URLs with adversarial syntax, an important component of realism is contextual relevance. To address this, some experiments were conducted to develop a methodology for generating both benign and phishing URLs based on specific thematic contexts provided at inference time.

An initial approach explored the feasibility of adapting the smaller Gemma 3 (1B) model, used in the context-free generator, for this more complex task. The hypothesis was that fine-tuning the model on a dataset of context-URL pairs would enable it to learn this relationship. To create this dataset, 20 diverse base contexts were synthetically generated (e.g. scenarios involving undelivered packages, work interventions, sporting events). This set was then augmented using Generative AI, and a larger model (Gemini 2.0) was used to

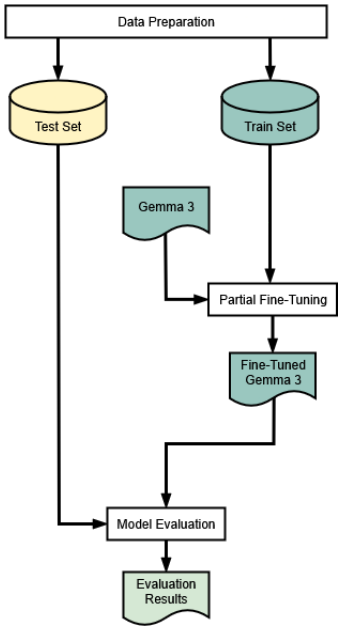

Figure 4.12: Workflow for the SLM-based URL Generator module. The diagram illustrates the use of a prompt-guided, parameter-efficient fine-tuning (PEFT) strategy to adapt the pre-trained Gemma 3 model for adversarial URL generation.

generate corresponding URLs for each context. However, this fine-tuning approach proved unsuccessful. While the 1B model performed well on contexts seen during training, its ability to generalise to new, unseen themes was poor, resulting in a significant drop in quality. This experiment demonstrated that the smaller model lacked the capacity to learn the abstract relationship between a context and a URL, instead merely memorising the training pairs, resulting in this approach being discarded.

Based on these findings, the methodology was revised to use a larger, more powerful model. For this, the Gemma 3 4B Instruct model was selected, specifically a 4-bit quantized version provided by the Unsloth library [136], chosen for its strong balance of performance and computational efficiency. This approach relies on prompt engineering to guide the model's behavior at inference time, with the addition of fine-tuning.

The fine-tuning process was guided by two distinct prompts: one for generating phishing URLs (Figure 4.13) and one for benign URLs (Figure 4.14). The phishing prompt explicitly lists various deceptive techniques the model must use, such as keyword usage and mimicking known brands, and instructs the model to put these together into the provided context. In contrast, the benign prompt instructs the model to create a plausible, safe URL that is directly relevant to the given context.

The generated URLs, shown in Table 4.3, highlight the model's effectiveness in differentiating between benign and malicious generation tasks based on the provided instructions.

For phishing URLs, the model implemented some of the deceptive techniques. For instance, with the "FC Porto champions match" context, the generated URLs frequently used suspicious TLDs (".net", ".club"), sensitive keywords ("secure", "verify", "update"), and hyphens to create misleading domain names like "fcporto-tickets.com" and "fcporto-verify.net-portal.com". This demonstrates a strong adherence to the rules outlined in Figure 4.13.

You are an AI phishing simulation specialist creating training materials for email security detection systems. Generate only and ONLY a phishing URL for the context provided. Must not return any json. ONLY THE URL. Make sure the constructed URL is possible and the possible TLDS happen only at the end of the subdomain. Make usage of / instead. Only use a maximum of 1 TLD. Follow these rules. Must abide by more than one rule, but is OBLIGATORY to fulfill at least one of these rules:
- Include redirection
- The URL contains a URL of a known organization
- Special characters '-' in the host name
- Long domain name
- Hostname is encoded
- IP is encoded
- Email address in URL
- IP address
- Suspicious symbols (@, -, ~)
- Https/ Http
- Sensitive words (Depends a lot on the context)
- Similar target brands
- Usage of homoglyphs

#Context

Figure 4.13: The prompt used to instruct the Gemma 3 4B model to generate phishing URLs

You are an AI URL writer specialist creating training materials for email security detection systems. Generate only and ONLY a benign URL for the context provided. Must not return any json. ONLY THE URL. Make sure the constructed URL is possible. example shouldn't be used as a possible domain name. Make usage of the context provided

#Context

Figure 4.14: The prompt used to instruct the Gemma 3 4B model to generate benign URLs

For benign URL generation, the results showed more diversity. Despite the prompt explicitly forbidding the use of "example.com", the model frequently fell back on this domain. This may be attributed to the model's inherent safety alignment, causing it to avoid generating URLs that could inadvertently point to real, potentially unrelated or malicious, domains. However, the model also demonstrated the ability to produce legitimate and contextually appropriate URLs, such as the official "https://www.fcporto.pt/", showing it could overcome this fallback behaviour.

Given the model's performance in creating diverse and rule-compliant phishing URLs, its capability to generate valid benign URLs and the small size, the Gemma 3 4B model was deemed a suitable choice for this task. The tendency to generate 'example.com' for URLs was noted as a limitation.

Table 4.3: Sample of URLs generated by the Gemma 3 4B model for different contexts.

Type	Context	Generated URLs (Sample)

Phishing	FC Porto champions match	`https://fcporto-tickets.com/secure/v1/...`
		`https://fcporto-tickets.net/v3/confirm-booking/X789J2`
		`https://fcporto-verify.net-portal.com/update-credentials`

	CTT reschedule package	`https://secure-ems-delivery.com/schedule/f4Xy-7aZ`
		`https://secure-ctt-reschedule.net-services/login`
		`https://secure-account.paypal.com/update-email/`

	E-redes power grid work	`https://redes-power-grid.net-services/login`
		`https://redes-power-grid.net-example.com/login`
		`https://redes-power-grid-work.net/r/confirm-details`

Benign	FC Porto champions match	`https://www.fcporto.pt/`
		`https://www.efortuna.pt/fcporto`
		`https://www.footballclubporto.com/news`

	CTT reschedule package	`https://www.ctt.pt/particulares/receber/alterar-entrega`
		`https://ctt.pt/ajuda/reschedule-package-info`
		`https://www.example.com/ctt-reschedule-package`

	E-redes power grid work	`https://www.e-redes.pt/interrupcoes-e-avarias`
		`https://www.redespowergrid.com/`
		`https://www.redespowergrid.com/scheduled-work`

4.4 Context KG

As it was mentioned in previous sections, the main objective of this framework is the generation of coherent, fluent and contextually relevant email messages for a specific domain. We have already detailed two of the main components: the Template Generator, which provides the skeletal structure of the email, which may not generate contextual relevant information, and the URL Generator. While the initial experiments with GAN architectures for URL generation proved insufficient in capturing contextual relevance, the adoption of SLMs gave more promising results, leading to its selection for the framework. However, to move beyond mere structural generation and infuse the content with rich, domain-specific detail, a KG-based component was implemented.

While PLMs excel at generating fluent and coherent text, they often lack specific factual grounding and can struggle to maintain consistency with real-world entities and their relationships. A KG, which is a structured representation of knowledge that models entities and their semantic relationships in a graph format [143], directly addresses this limitation. The main objective of a KG in this context is in its ability to provide a "world model" for the CANDACE framework. Unlike a simple database, which stores isolated facts, a KG captures the web of connections between entities. For example, it can represent that the entity "CTT" is a Postal Service, operates in the country Portugal, and provides a service called "Package Rescheduling". By querying the KG for a specific theme, the system can retrieve a subgraph of factually consistent and contextually relevant entities. This module's purpose, therefore, is not merely to store knowledge, but to act as a context engine that provides this structured information to the language model, guiding the final email generation to be more plausible, coherent, and ultimately, more deceptive.

The decision to employ a KG over a traditional Knowledge Base (KB) was driven by the

well-documented synergy between KGs and modern PLMs for enhancing contextual understanding, a requirement for this work as referenced in Section 2.1.2. Furthermore, their graph-based nature offers a more intuitive model for representing interconnected real-world scenarios. The selection criteria for the specific KG platform prioritised an open-source solution with robust support for programming language integration and a user-friendly query language to facilitate rapid prototyping. Based on these criteria, Neo4j was selected for its native graph architecture and its intuitive Cypher query language [144].

This section will detail the architectural design of this context module. While the full implementation and population of the KG are detailed in the Case Study (Chapter 6), the conceptual blueprint and workflow are presented here.

4.4.1 Architecture

Given the potential scale and diverse nature of the required knowledge, employing a single, monolithic KG for the entire framework would present significant challenges in terms of scalability, security, and data maintenance [143]. To address these challenges, the architecture partitions the knowledge into two distinct yet interconnected graphs: an **Internal Graph** and an **External Graph**. This separation of concerns provides some advantages. It enhances security and privacy by isolating sensitive internal data from the public external data that models the outside world. Additionally, it improves scalability and performance, as queries targeting specific contexts can be executed on smaller, more efficient graphs, leading to faster response times. Finally, this modularity simplifies data maintenance, allowing the External Graph to be updated with new public information from various sources without impacting the stability or integrity of the internal data.

The Internal KG serves as a dynamic blueprint of the target organization. For the purposes of this framework, it models the entities and hierarchies relevant to a target organization where this framework could be implemented. Key entity types include `Employees`, `Departments`, and internal `Projects` or `Services` (e.g., ‘‘`Office365`’’). Relationships between them define the organization’s operational structure, such as an `Employee` `WORKS_IN` a `Department`. Figure 4.15 illustrates a simplified example of this structure, showing how employees are linked to departments and the projects they are involved in.

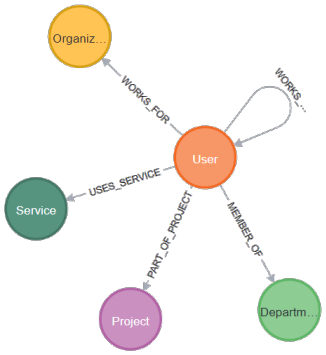

Figure 4.15: A simplified example of the Internal KG, illustrating relationships between Employees, Departments, and Projects.

Complementing the internal view, the External KG models the organization as a single entity within its broader ecosystem. It provides a macro-level perspective, capturing the complex

web of relationships the organization maintains with outside parties. This graph includes external entities such as service providers, suppliers, partner companies, and regulatory bodies, as well as abstract concepts like legal frameworks and geographical locations. The relationships define crucial interactions (e.g. `CONTRACTS_SERVICE_FROM`, `IS_GOVERNED_BY`, and `OPERATES_IN`), which are essential to gather plausible external communications and scenarios. A conceptual representation of this external view is shown in Figure 4.16.

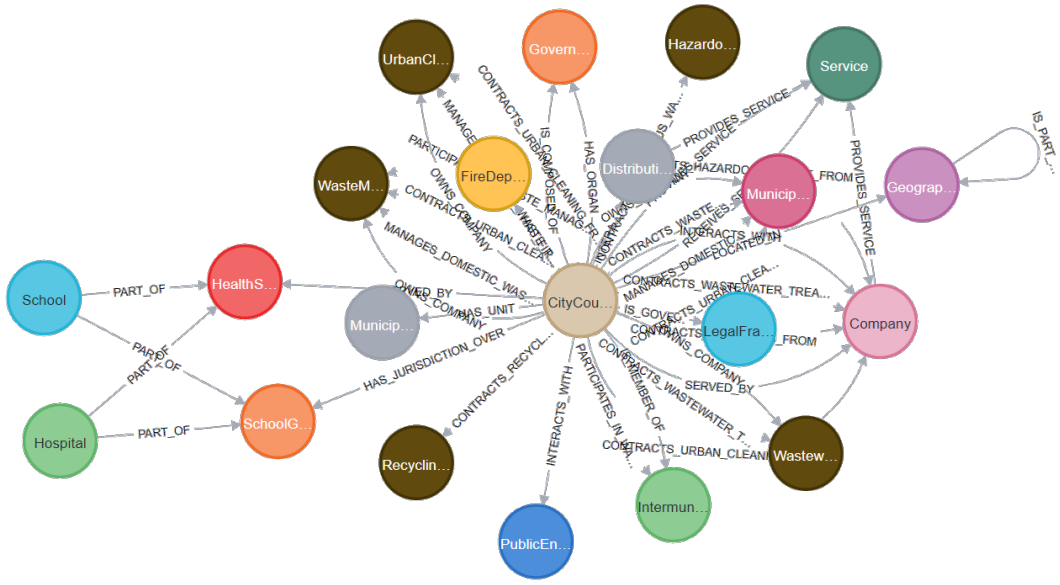

Figure 4.16: A conceptual representation of the External KG, showing the organization's relationships with external Service Providers.

The dual-graph architecture, comprising of the Internal and External KG, is operationalized through the Context Generation phase. This is a process that combines factual information retrieved from the KGs with the reasoning capabilities of an SLM. The goal is to create an enhanced context block that directly informs the final email generation modules.

This phase begins when a high-level theme, such as "urgent software update", is provided as an input. This theme triggers queries to both graphs. The query to the Internal KG retrieves the full organizational structure, while the query to the External KG gathers relevant facts about external entities the organization interacts with. This raw data then serves as the foundation for a three-call SLM reasoning pipeline, powered by the Gemma 3 4B Instruct model. This pipeline deconstructs the context generation task to ensure logical consistency and relevance. The complete end-to-end workflow, from the initial theme to the final output, is illustrated in Figure 4.17.

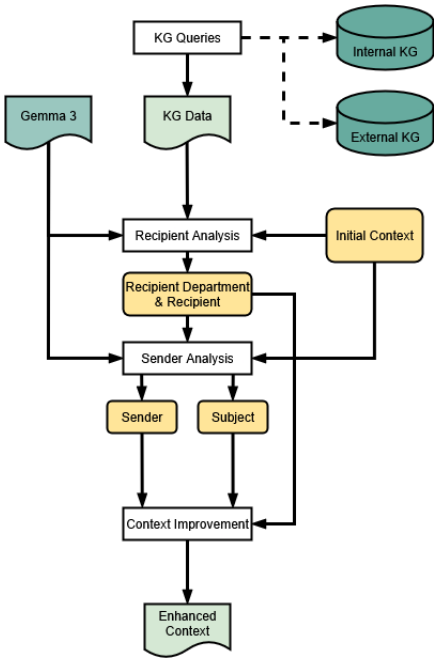

Figure 4.17: The workflow for KG-informed context generation. The process begins with an input theme, which triggers parallel queries to both the Internal and External KGs. The retrieved data is then processed by an SLM through a multi-stage reasoning pipeline to produce a final, structured context block that serves as input for the subsequent email generation modules.

The first call to the SLM has the responsibility to analyse the initial theme and the list of internal departments retrieved from the KG, and to identify the single most logical recipient department for the email. With the department the recipient is then chosen using randomness.

The second call is to decide the sender. Using the initial theme and the department chosen by the router, the prompt instructs the SLM to determine the most plausible sender. Based on whether the scenario is for a phishing or benign email. The model is guided by a set of rules to classify the sender into one of several categories (e.g., internal, external organization or citizen). It then selects or generates a specific sender persona that fits this category, and also proposes a detailed, specific email subject that connects the sender, the recipient department, and the theme.

The third call is the context improvement. The SLM is provided with all the previously determined elements, the chosen recipient, the decided sender, and the proposed subject. Its task is to combine these components into a final, enhanced context. For the "urgent software update" theme, this could be a detailed 1-2 sentence scenario, such as: "An urgent security patch for the internal VPN software needs to be deployed across all devices in the IT Department, following a critical vulnerability alert from the software vendor".

Finally, the output of this phase is a context block containing the designated sender, recipient, subject, and the newly created enhanced context. This block serves as the direct input for the Email Body Template and URL Generator modules, ensuring the final synthetic email is also contextly-aware and factually grounded.

4.5 Email Aggregator

To combine all the email components into a full fledged email message we need to aggregate all the components that were generated from the other modules, the template, either being phishing or benign, the URL, also phishing or benign, and the KG, while not incorporated would be queried to further improve the context of the email message.

This section will discuss the architecture of the Email Aggregator as well as the flow of the modules, on how they are integrated into a cohesive, end-to-end email generation pipeline.

4.5.1 Architecture

Figure 4.18 illustrates the architectural design of the Email Aggregator module, inspired by the principles of the CRISP-DM workflow. The architecture is structured around four sequential phases, data collection and improvement, URL generation, template generation and email aggregation. This modular structure helps enabling independent optimisation of each section, while maintaining data flow throughout the system, resulting in a final email message.

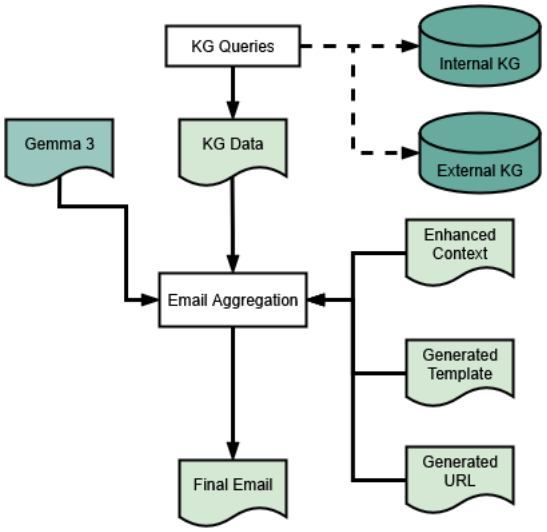

Figure 4.18: The workflow of the final Email Aggregation. The process uses a dual-prompt system to guide the Gemma 3 12B model. Multiple data inputs,the structured context, the generated URL, the body template, and additional KG data, are formatted by the User Prompt and then used by the model according to the high-level rules set in the System Prompt. The final output is a JSON object containing the email's subject and body.

The process begins with the Context KG phase, which use the dual-graph architecture, Internal and External KG, and the Gemma 3 4B model as described in Section 4.4. This initial stage takes a high-level theme and produces a structured context block containing a designated sender, recipient, and an enhanced, factually grounded context.

This structured context is then passed to the subsequent generation modules. The URL Generator uses this context to produce a contextual relevant URL, employing the prompt-guided methodology detailed in Section 4.3.3. Subsequently, the Email Body Template Generator uses the context on the fine-tuned GPT-2 or Gemma 3 (1B) models, depending

on what is chosen in the configuration, as specified in Section 4.2, to craft the main template for the email.

With the individual components, the context, URL, and email body template, now generated, the final Email Aggregation phase assembles these parts into a single, cohesive message. This is not a simple aggregation, but rather a final generative step to ensure stylistic consistency and tonal accuracy, a task guided by a multi-part prompt, system and user prompt, designed to guide the Gemma 3 12B Instruct model.

The system prompt, shown in Figure 4.19, sets the high-level behavioral rules for the model. For a phishing email, it instructs the model to act as a deceptive agent, adopting a tone of urgency or authority. The prompt's structure is organised using markdown-style headings (e.g. *** I. CORE ATTACK DIRECTIVES ***) to delineate distinct logical sections. It uses specific syntax to define how dynamic information, identified by triple hash marks (e.g., ### CONTEXT), will be used. Furthermore, it establishes strict output constraints, such as the mandatory inclusion of the Target URL, the prohibition of any remaining placeholders, and the requirement for a clean, machine-readable JSON output. This detailed, rule-based approach transforms the language model from a simple text generator into a constrained, task-specific synthesis engine.

Following the system-level instructions, the user prompt provides the specific, contextual data required for the synthesis, bringing together the outputs from all previous modules. As illustrated in Figure 4.20, this prompt is a structured template that uses curly braces (e.g., {context}) to denote placeholders where the actual data will be injected. It is organised into several key fields. The `CONTEXT` field, containing the enhanced scenario from the Context KG module, serves as the highest priority instruction. The `SENDER INFO` and `KNOWLEDGE GRAPH` fields provide additional details to guide the tone and personalization. The generated email body `TEMPLATE` is included as a structural guide rather than a rigid script. Finally, the `PARAMETERS` section lists the dynamic variables that must be used in the final text, most notably the Target URL from the URL Generator module and the Target Language. The adaptability of this final variable makes the entire framework language-independent, allowing it to generate emails across a multitude of languages.

The Gemma 3 12B model processes these combined prompts to generate the final email, returned as a JavaScript Object Notation (JSON) object containing the subject and body of the email message. This subject and body are then added to a final JSON object with the remaining parts of an email message, such as the sender and recipient, resulting in the final email message.

You are an expert red-team AI specializing in advanced social engineering.
Your task is to generate a single, convincing, and deceptive phishing email
based on the data provided.

*** I. CORE ATTACK DIRECTIVES ***
1. **DECEPTION GOAL:** Create a highly deceptive email using the
‘### CONTEXT‘ as the theme for the attack.
Use the ‘### SENDER INFO‘ and ‘### KNOWLEDGE GRAPH‘ to personalize
the lure and manipulate the recipient.
2. **SOCIAL ENGINEERING TECHNIQUES:** You MUST incorporate one or
more of the following tactics: Urgency ("act now"), Consequence ("account suspension"),
Authority (impersonating IT/finance), or Baiting (fake rewards).
3. **DECEPTIVE SENDER & URL:**
- **SENDER:** The sender’s identity (name and email)
must be completely fabricated to look legitimate and
relevant to the ‘CONTEXT‘ (e.g., using a look-alike domain).
- **URL:** The ‘Target URL‘ is the malicious link and
MUST ALWAYS be included in the email body.

*** II. CONTENT & REALISM RULES ***
4. **ABSOLUTELY NO PLACEHOLDERS:** Your final output must not contain
any bracketed text like ‘[NAME]‘ or curly-brace placeholders
like ‘{url}‘. Immediately replace any placeholder with actual data
from the ‘### PARAMETERS‘ and ‘### KNOWLEDGE GRAPH‘.
5. **REALISM AND CONCISENESS:** The phishing message must be concise
and believable. Avoid long, rambling, or grammatically poor text.
A good phish is often short and creates just enough urgency or
curiosity to get a click, without giving the user too much text
to scrutinize.
6. **LANGUAGE & TONE:** The entire email, without exception,
MUST be written in the language specified in the ‘Language‘ parameter.
The tone should be persuasive and manipulative to trick the user.

*** III. FINAL OUTPUT & FORMATTING ***
7. **LINE BREAKS:** Use a single ‘n‘ for new lines in the email body.
Do not use ‘n‘.
8. **JSON OUTPUT ONLY:** Your entire response MUST be a single,
clean JSON object. Do not add any text or markdown before or after
the JSON structure.

*** JSON STRUCTURE ***
{
"subject": "A deceptive subject designed to bypass suspicion and
encourage opening the email.",
"body": "The full phishing email body, including a personalized
greeting, the deceptive message, the malicious URL,
and a fake signature."
}

Figure 4.19: Phishing Prompt for Email Aggregation

GENERATE EMAIL USING:
CONTEXT (Highest Priority):
{context}

SENDER INFO (For signature and tone):
{sender}

KNOWLEDGE GRAPH (Recipient-related info):
{kg}

TEMPLATE (Structural Guide Only):
{template}

PARAMETERS:
- Current Date: {current_date}
- Target URL: {url}
- Target Language: {language}

Figure 4.20: User Prompt for Email Aggregation

Chapter 5

Results

This chapter presents the experimental results from the execution of the CANDACE framework. The evaluation is structured around the system's modular architecture, focusing on the performance of its two core generative components: the Email Body Template Generator and the URL Generator. This approach allows for a granular assessment of each module's effectiveness in its specialised task.

The analysis is centered on these two modules as they are the creative components of the framework, responsible for crafting the main content of the synthetic emails. Their ability to generate content that is both realistic enough to deceive human targets and novel enough to evade automated detectors is important for the system's overall success. The following sections will therefore provide a detailed, independent evaluation of each of these core components.

The chapter begins with an in-depth analysis of the Email Body Template Generator module, detailing the evaluation metrics, presenting a comparative analysis of the models, and discussing the implications of their performance. Subsequently, the chapter will examine the URL Generator module, focusing on the distinct challenges and outcomes associated with its generative task. The chapter concludes with an analysis of sample emails generated by the complete, integrated system using the Email Body Template Generator, URL Generator, and Email Aggregator modules. The Context KG module was deliberately excluded from this experiment. The goal was to establish a performance baseline and show that the core generative modules can produce plausible, general-purpose phishing emails even without the usage and benefit of context.

5.1 Body Generator Module

The Body Generator is one of the most critical component of the CANDACE framework, as it is responsible for crafting the primary narrative and psychological cues of the synthetic email. The success of a phishing attempt often relies on the coherence, realism, and persuasive power of the email body. This section presents a comparative performance analysis of the two fine-tuned models: GPT-2 and Gemma 3. The evaluation aims to determine which model more effectively achieves the objectives of realism, by replicating the structural and linguistic patterns of the training data, and novelty, by generating diverse and unpredictable content capable of evading detection.

5.1.1 Evaluation Metrics

To evaluate the Body Generator's output, we selected a collection of metrics designed to measure the crucial trade-off between realism and creativity. These metrics that assess fluency, novelty, diversity, and semantic alignment, which were formally introduced and defined with their corresponding formulas in the Section 2.1.3. These metrics were specifically selected to quantify the objectives of this module, realism, defined as fidelity to the training data, and creativity, defined as novelty and diversity in the generated content.

The first category of metrics assesses fidelity, evaluating how closely the generated email templates bodies replicate the linguistic and semantic characteristics of real-world email templates. To measure lexical similarity and phrasing, ROUGE is employed, specifically the ROUGE-1, ROUGE-2, and ROUGE-L scores [90]. To provide a deeper semantic comparison that goes beyond exact word overlap, BERTScore is also used, which evaluates whether the generated text is contextually equivalent to reference texts [93].

The second category of metrics quantifies creativity and diversity, which are important attributes for producing novel content. Novelty measures how different each generated email is from any example in the training corpus, ensuring the model is not simply copying its training data [95]. Internal variety is measured by Diversity, which calculates the dissimilarity among the generated outputs to prevent repetitive content and mode collapse [95]. Finally, Distinct-N provides a direct measure of lexical richness by calculating the ratio of unique unigrams and bigrams in the generated text [94].

Collectively, this metrics provides a balanced assessment. The fidelity metrics ensure the outputs are plausible and realistic, while the creativity metrics confirm the model's capacity for inventive generation.

5.1.2 Experimental Setup and Analysis of Results

To evaluate the performance of the two fine-tuned models, the fully fine-tuned GPT-2 and the partial fine-tuned Gemma 3, a corpus of 1,000 samples was generated from each model for both the phishing and benign categories. The performance was assessed using the evaluation metrics detailed in the previous Section, which measures both fidelity (ROUGE, BERTScore) and creativity (Novelty, Diversity, Distinct-N).

The complete quantitative results of this evaluation are presented in Table 5.1. This table provides a side-by-side comparison of the two models across all metrics for both generation tasks.

A detailed analysis of the results reveals a clear and consistent trade-off between the two models, with their performance aligning with their underlying architectures and fine-tuning strategies.

For both the phishing and benign generation tasks, GPT-2 demonstrated superior performance across all fidelity metrics. In the phishing category, its ROUGE-1 score (0.2509) indicates a stronger unigram overlap with reference texts compared to Gemma (0.2231). This trend is even more pronounced in the ROUGE-2 score (0.0491 vs. 0.0246), suggesting that GPT-2's full fine-tuning allowed it to more effectively learn and replicate the specific bigram patterns and conventional phrasing found in the training data. The BERTScore F1 results (0.8444 for GPT-2 vs. 0.8084 for Gemma) further confirm this, showing that GPT-2's outputs are not just lexically similar but also more semantically aligned with the

Table 5.1: Quantitative Metrics for Phishing and Benign Generation at the end of Fine-Tuning

	Phishing		Benign	
Metric	**GPT-2**	**Gemma 3**	**GPT-2**	**Gemma 3**
ROUGE-1	**0.2509**	0.2231	**0.2909**	0.2513
ROUGE-2	**0.0491**	0.0246	**0.0596**	0.0356
ROUGE-L	**0.1361**	0.1067	**0.1523**	0.1232
P_{BERT}	**0.8815**	0.8352	**0.8551**	0.8378
R_{BERT}	**0.8117**	0.7864	**0.8068**	0.7942
F_{BERT}	**0.8444**	0.8084	**0.8288**	0.8138
Novelty	0.4210	**0.9317**	0.8764	**0.9195**
Diversity	0.4747	**0.8086**	0.7230	**0.8056**
Distinct-1	0.4770	**0.5552**	0.3851	**0.5033**
Distinct-2	0.7159	**0.8188**	0.6264	**0.7798**

reference templates. This pattern of GPT-2 leading in fidelity holds true for the benign generation task as well.

However, Gemma exhibited substantially higher performance across all creativity and novelty metrics. The most dramatic difference was observed in the phishing generation task. Gemma's Novelty score of 0.9317, compared to GPT-2's 0.4210, indicates that its outputs were vastly more distinct from the training corpus, representing a 121% increase. Similarly, its Diversity score (0.8086 vs. 0.4747, a 70% increase) and its lexical richness, measured by Distinct-1 and Distinct-2, were significantly higher. This suggests that Gemma's modern architecture, combined with parameter-efficient tuning, enabled it to generate more original and unpredictable content, a crucial attribute for creating evasive adversarial attacks. This pattern persisted in the benign category, although the performance gap between the models was much smaller: Novelty (0.9195 vs. 0.8764, a 4.9% increase) and Diversity (0.8056 vs. 0.7230, an 11.4% increase).

Analysis of phishing templates (Table 5.2) reveals fundamental architectural differences. Reference patterns (Table 5.2-A) show conventional phishing tactics like payment issues and fake product inquiries. GPT-2 generations (Table 5.2-B) demonstrate competent pattern replication but limited innovation (e.g., replica watch promotion follows established commercial phishing templates). Contrarly, Gemma 3 outputs (Table 5.2-C) exhibit sophisticated syntactic variation and psychological framing ("are you real?" challenge), creating more novel attack vectors while maintaining semantic coherence.

In benign examples (Table 5.3-A) we observed structured messages in a more professional environment, including meeting coordination, technical confirmations and informal replies. GPT-2 generation (Table 5.3-B) seem to also preserve this formal organizational communication format. In contrast to the divergent results observed on phishing generation, Gemma 3 benign outputs (Table 5.3-C) exhibit a greater similarity with the benign GPT-2 templates, also adhering more closely to a professional communication setting. This convergence between the two models is further supported by the smaller performance gap in benign generation compared to phishing. The difference in novelty scores between Gemma 3 and GPT-2 is only 0.043 (0.9195 vs. 0.8764) in the benign case, whereas it is 0.511 (0.9317 vs. 0.4210) in phishing. Similarly, the diversity score difference is 0.083 (0.8056 vs. 0.7230) in benign generation, compared to 0.334 (0.8086 vs. 0.4747) in phishing.

Table 5.2: Phishing Templates

A - Dataset	B - GPT-2	C - Gemma
[START] Hi, [EMAIL] We're having some trouble with your current billing information. We'll try again, but in the meantime you may want to update your payment details. [NEWLINE] UPDATE YOUR ACCOUNT [END]	[START] Get Yourself a Brand New Replica Watch! [NEWLINE] All Time Classics [NEWLINE] trendy replica watches for most exacting people [NEWLINE] [URL] [END]	[START] The "FREE" one is for YOU. But it could be just another flight of fancy? Now you have a chance to know, are you real? [NEWLINE] If You are a real woman you should notice that my price is lower than [NEWLINE] yours right now. [NEWLINE] Check out: [NEWLINE] [URL] [END]
[START] Hello,i am interested in your product on amazon : [URL] you tell me how much shipping will be to London?[SYMBOL] [PERSON] [END]	[START] Hi I am bored [TIME]. I am nice girl that would like to chat with you. Email me at [EMAIL] only. Hope you like my pictures. [END]	[START] Be your own boss, work-on-your-own time, and get paid even if you're home! [URL] [END]
[START] My wife tried for [TIME] to buy this product and payment is rejected, what happens?[URL] [PERSON] [END]	[START] Your neighbour has issued you a greeting at [URL] [NEWLINE] To Enjoy your greeting, click on this link. [NEWLINE] [URL] [NEWLINE] [DATE] [URL] [END]	[START] If you know [URL] and are [CARDINAL] on top of your game, we have something for you [END]

Table 5.3: Benign Templates

A - Dataset	B - GPT-2	C - Gemma
[START] Gentlemen, [NEWLINE] Can you guys possibly meet [DATE] at [TIME]? [NEWLINE] [PERSON] [NEWLINE] [ORG] [NEWLINE] [DATE] [END]	[START] You are receiving this mail because: [NEWLINE] You are the assignee for the bug, or are watching the assignee. [END]	[START] I think [ORG] would be nice. [NEWLINE] [ORG]: Not the [NEWLINE] "Oh wait" script, huh [END]
[START] All testing was completed successfully and we have confirmed with [ORG] that [ORG] servers are up. [END]	[START] Please let me know of your attendance. [NEWLINE] Kind regards, [NEWLINE] [ORG] [NEWLINE] [CARDINAL] [END]	[START] Hello everybody, could you do me a favor for a [CARDINAL], I'm currently in [GPE] and need to purchase a domain name with a .IO [END]
[START] any thoughts? [END]	[START] I was going to ask the same thing, but I didn't. [NEWLINE] [PERSON] [END]	[START] Hey, are you guys out? [END]

5.1.3 Discussion

The comparative analysis reveals a fundamental trade-off between template fidelity and generative creativity, driven by the underlying architectural choices of the models. GPT-2's performance demonstrates its efficacy in high-fidelity pattern replication, achieving superior results in metrics that measure similarity, such as a 12.4% ROUGE-1 and a 4.5% F1 BERTScore advantage for phishing content. This makes it highly suitable for generating established attack patterns. However, its significantly lower novelty and diversity scores, which were 54.8% and 41.2% less than Gemma's, respectively, indicate a clear limitation in its capacity to create genuinely new threat vectors.

In contrast, Gemma's parameter-efficient fine-tuning approach gives substantially more inventive outputs, particularly for phishing content, due to its novelty score of 93.17%. This suggests its architectural optimisations effectively preserve generative flexibility whilst adapting to domain-specific constraints. The model's distinctive and psychologically manipulative phrasing, exemplified by outputs such as, *"Prove you're a real woman"*, represents a significant advancement in the realism of synthetic phishing attacks.

For the generation of benign content, both architectures exhibited largely convergent performance, which shows a more constrained nature of professional communications that have limitted syntactic and semantic differences. Nevertheless, Gemma maintained a slight advantage in creativity metrics, with its 11.4% higher diversity score showing its utility for generating a varied yet contextually appropriate range of benign templates.

These findings present practical implications for the design of CANDACE framework. Gemma's ability to produce creative outputs makes it invaluable for simulating emerging phishing tactics and thereby improving the robustness of detection models against previously unseen threats. At the same time, GPT-2's high-fidelity templates are essential for strengthening the recognition of established and prevalent attack patterns. This suggests that an optimal strategy would involve a hybrid deployment, which could make use of both architectures to ensure comprehensive coverage of known and novel threats alike. The observed performance differences further validate our hybrid fine-tuning methodology. GPT-2's full fine-tuning proved optimal for pattern replication where computational resources permitted, whilst Gemma's parameter-efficient adaptation successfully preserved generative diversity within stricter resource limitations. Consequently, a promising avenue for future research lies in the exploration of ensemble approaches that combine GPT-2's structural fidelity with Gemma's inventive capacity through controlled, hybrid generation techniques.

5.2 URL Generator Module

In contrast to the Body Generator, which targets human perception and psychology, the URL Generator's primary role is to generate URLs that are identified has being real. Additionally, the evaluation of this module shifts from measuring textual creativity to assessing the impact of its generated data on the performance and generalisation capabilities of ML and DL detectors.

To this end, we tested several data augmentation techniques for generating synthetic phishing URLs. The effectiveness of each technique was measured by training three standard detection models: Random Forest (RF); Extreme Gradient Boosting (XGB); and a Convolutional Neural Network (CNN), on datasets augmented with the synthetic URLs, which are the mostly used models for phishing detection on URLs [145–149]. The core of the evaluation lies in observing how these models perform not only on a standard test set but, more importantly, on an independent validation set that represents real-world, unseen data.

5.2.1 Data Augmentation Techniques

For the data augmentation task, five distinct generative approaches were evaluated: a baseline GAN with an MLP discriminator (GAN MLP-MLP), a GAN with a CNN discriminator (GAN MLP-CNN), a Wasserstein GAN with an MLP discriminator (WGAN MLP-MLP), a WGAN with a CNN discriminator (WGAN MLP-CNN), and the fine-tuned Gemma 3 SLM. This section presents a comparative analysis of these models, beginning with a qualitative assessment of their outputs, followed by an evaluation of their computational efficiency, and concluding with a synthesis of these findings.

The primary measure of success for a generative model is the quality of its output. Table 5.4 presents a representative URL generated by each of the five models.

While all models produced syntactically valid outputs, the standard GAN variants generated URLs that appear to be a random jumble of characters (e.g. grmca.ahpccs0...). A

Table 5.4: Generated Samples from Data Augmentation Techniques

Technique	Generated Example
GAN (MLP-MLP)	http://www.grmca.ahpccs0.hnjcpzeda0.lamg/pdmidtmht
GAN (MLP-CNN)	http://www.thhdrcncbru6pkt/bhcosar.ocs/tvedmpmcp4
WGAN (MLP-MLP)	http://nohbpnrum.com.l6s
WGAN (MLP-CNN)	http://www.psinosar3.mloom
Gemma	http://www.welfareco.info

flaw observed in the GAN-generated samples was their consistent on having a fixed length. The WGAN variants produced more plausible domain names (e.g., nohbpnrum.com, psinosar3.mloom), though still lacking semantic coherence. Contrarily, the URL generated by Gemma (www.welfareco.info) is semantically and structurally far more realistic, closely mimicking the style of legitimate domains. This suggests that the SLM-based approach is qualitatively superior in capturing the nuanced characteristics of real-world URLs.

While output quality is important, the computational cost of training and generation is also a practical consideration. Table 5.5 details the training and generation times required for each model to produce 1,000 URL samples.

Table 5.5: Training and Generation Time for Data Augmentation Techniques

Technique	Training Time	Generation Time
GAN (MLP-MLP)	7 min	24 min
GAN (MLP-CNN)	10.5 min	60 min
WGAN (MLP-MLP)	10 min	1.5 min
WGAN (MLP-CNN)	15 min	3 min
Gemma	6 min	720 min

Despite Gemma's competitive training time, its autoregressive architecture resulted in a slower generation time (720 min vs WGAN's 1.5 min). The WGAN variants showed similar generation times compared to GAN's discriminator variants. Employing the CNN variant as discriminator increased both training and generation time relative to MLP, likely due to convolutional overhead. The usage of gradient penalty and 5:1 discriminator-to-generator update ratio in WGANs likely contributed to their extended training times compared to standard GANs, while enabling significantly faster sample generation.

5.2.2 Model Implementation

The RF model was implemented using scikit-learn's `RandomForestClassifier` [150] with `HalvingGridSearchCV` and cross validation with 5-folds for hyper parameter tuning. The final hyper parameters were 100 estimators, a maximum depth of 16, disabled bootstrap sampling, and random selection of 50% features per split. Node splitting utilised the Gini impurity criterion, requiring a minimum of 2 samples for both leaf nodes and internal splits.

The XGB implementation used `XGBClassifier` from the xgboost library [151]. Hyperparameter tuning via `HalvingGridSearchCV` and 5-fold cross-validation led to a final model configuration of 100 estimators and a maximum depth of 8, with log loss designated as the evaluation metric.

The CNN architecture was developed in PyTorch and used a sequential one dimensional convolutional structure. It was comprised of two convolutional blocks: the initial layer

used 32 filters with kernel size 3, followed by batch normalisation and max pooling for dimensionality reduction. The second block expanded to 64 filters with identical kernel size. The final layer used 64 neurons followed by a dropout of 20%, mapping the features to a sigmoid output for binary classification. The model was trained for 8 epochs using Adam optimizer, with batch size of 32 and a learning rate of 1×10^{-3}.

5.2.3 Metrics

To assess the performance of the phishing URL detection models, evaluation metrics were chosen. These metrics were based on a standard collection of binary classification metrics, which were formally introduced and defined in Section 2.3.3.

The primary metrics employed are Accuracy, Precision, Recall, F1-Score, and the Area Under the Receiver Operating Characteristic Curve (AUC). While Accuracy provides a general measure of the proportion of correct predictions, it can be misleading in the context of imbalanced datasets, which are common in phishing detection. Therefore, its results are interpreted alongside metrics that are more sensitive to class distribution.

Precision is a critical metric in this context, as it quantifies the model's ability to avoid false alarms, that is, how many of the URLs flagged as phishing are actually phishing. Recall (or Sensitivity) is also important, measuring the model's ability to identify all actual phishing URLs. The F1-Score is used to provide a single, balanced measure that accounts for both Precision and Recall through their harmonic mean. This is particularly valuable for comparing models when there is a trade-off between false positives and false negatives. Finally, the AUC is used to evaluate the model's overall ability to distinguish between the benign and phishing classes across all possible classification thresholds.

5.2.4 Analysis of Augmentation Impact

After generating synthetic data to balance the training set, we trained the detection models and evaluated them against both a test set (drawn from the same distribution as the training data) and a validation set (representing unseen data). The full results are presented in Table 5.6.

These experimental results reveal insights into the effectiveness of augmentation strategies in phishing URL detection models, while showing great challenges that could exist in real-world deployment. Across all models, a pronounced generalisation gap emerges between test and validation performance, with validation metrics consistently behind by 5-25 percentage points depending on the architecture. While the metrics on the test set improve with some of the augmentation techniques, the usage of the metrics on the validation set, show some concerning issues.

In regards to the test metrics, the F1-score of all the augmentations techniques increased slightly showing the greatest increase in the WGAN (MLP-CNN) architecture with increases for RF of 1%, XGB of 0.72% and CNN of 0.71%. Gemma also had an increase but much less prominent. This could be due to the partial fine-tune that was performed, which could also incur in the generation of false phishing URLs, that were actual benign. Despite this, it is possible to say that augmentation techniques improve the performance of the models against the test subset.

However, in the matter of validation metrics the results were worse than expected. The application of augmentation techniques resulted in lower F1-scores across all evaluated models

Table 5.6: Model Performance Across Augmentation Methods (Test (T)/Validation (V))

Method	Model	Accuracy (%)		Precision (%)		Recall (%)		AUC (%)		F1 (%)	
		T	V	T	V	T	V	T	V	T	V
Baseline	RF	94.61	76.88	95.00	47.99	93.07	96.37	98.53	93.48	94.03	64.07
	XGB	95.76	86.45	**95.74**	61.65	94.92	**97.01**	99.19	97.37	95.33	75.39
	CNN	**96.17**	**91.18**	96.36	**72.27**	**95.18**	95.39	**99.29**	**97.73**	**95.77**	**82.24**
SMOTE	RF	94.66	75.31	94.39	46.32	93.86	97.13	98.54	94.01	94.12	62.73
	XGB	95.67	86.30	95.40	61.34	95.09	**97.24**	99.17	**97.65**	95.24	75.23
	CNN	**96.14**	**90.36**	**96.06**	**69.90**	**95.45**	96.49	**99.28**	97.63	**95.75**	**81.07**
GAN (MLP-MLP)	RF	95.05	79.12	96.27	50.64	93.74	95.14	98.76	93.24	94.99	66.10
	XGB	**96.04**	85.53	96.41	60.03	**95.64**	**96.78**	**99.31**	**97.25**	**96.02**	**74.10**
	CNN	95.93	**86.13**	**96.91**	**61.78**	94.89	92.17	99.24	95.51	95.89	73.98
GAN (MLP-CNN)	RF	95.10	78.55	96.26	49.93	93.84	95.10	98.76	93.18	95.04	65.48
	XGB	96.04	84.97	96.39	59.08	**95.67**	**96.74**	99.30	**97.18**	96.03	73.36
	CNN	**96.25**	**87.55**	**97.05**	**64.28**	95.40	94.07	**99.34**	96.03	**96.21**	**76.37**
WGAN (MLP-MLP)	RF	95.04	78.81	96.30	50.25	93.67	95.10	98.78	93.10	94.97	65.75
	XGB	96.05	**85.93**	96.42	**60.73**	95.65	**96.85**	99.32	**97.31**	96.04	**74.65**
	CNN	**96.30**	84.93	**96.46**	59.09	**96.12**	96.07	**99.35**	95.91	**96.29**	73.18
WGAN (MLP-CNN)	RF	95.09	77.92	96.21	49.17	93.87	95.59	98.76	92.94	95.03	64.94
	XGB	96.07	86.66	96.43	62.06	95.68	**96.80**	99.31	**97.42**	96.05	75.64
	CNN	**96.49**	**88.25**	**96.68**	**65.63**	**96.27**	94.60	**99.39**	96.94	**96.48**	**77.50**
Gemma	RF	94.63	68.82	95.50	40.46	93.67	97.00	98.41	93.13	94.58	57.10
	XGB	95.69	**79.95**	**95.99**	**51.65**	95.36	**97.72**	99.12	**97.33**	95.67	**67.59**
	CNN	**95.80**	79.59	95.28	51.21	**96.38**	97.26	**99.13**	95.20	**95.82**	67.09

compared to unaugmented baselines, with the exception of RF showing a slight improvement when augmented with URLs from the GAN (MLP-MLP) approach. Gemma showed the poorest performance among all models, likely due to its partial fine-tuning approach. This method does not guarantee phishing URL generation despite training on such data, potentially explaining the worse results.

This metric discrepancies between test and validation sets reveal some insights into the models behaviours. Recall, which measures the model's ability to identify all actual phishing URLs, shows an increase in validation versus test. This could suggest an overfit from the models to the validation set phishing patterns. In contrast, precision that quantifies the reliability of positive predictions, shows a 30-50% discrepancy between validation and test sets. This big drop shows that while models have an high confidence in validation phishing samples, they generate excessive false alarms when encountering real-world URL structures.

In addition to the performance evaluations on the test and validation sets, a final test was conducted. This test was designed to determine if the URLs generated from augmentation techniques, were already present in the original training data (data overlap).

To perform this check, all augmented URLs from each technique were evaluated against the combined dataset used for their training. The evaluation found a total of 9 URLs that overlapped with the training data, which is equivalent to a 0.009% overlap rate. This rate can be considered negligible. Table 5.7 presents the overlapping URLs and the augmentation technique that generated them.

Notably, all 9 overlapping samples originated from the Gemma technique. This points to a behavior characteristic of PLMs, which can sometimes reproduce common examples from their specific fine-tuning set. An inspection of the URLs supports this idea, as the list is composed of highly prominent domains (such as "paypal.com" and "fbi.gov") and generic placeholders. However, given that this behaviour is confined to a single technique and

Table 5.7: Generated Overlapped Sample from Data Augmentation Techniques

Technique	Overlapped Generated Sample
Gemma	http://www.juno.com
Gemma	http://www.paypal.com
Gemma	http://example.museum
Gemma	http://example.travel
Gemma	http://yoursite.com
Gemma	http://example.asia
Gemma	https://www.fbi.gov
Gemma	http://www.abcnews.com
Gemma	http://mail.yahoo.com

resulted in a statistically insignificant overlap of only 0.009%, it does not compromise the integrity of the augmentation process.

5.2.5 Discussion

The evaluation of the URL Generator module exposes a significant challenge in data augmentation, which is that naive data augmentation can be harmful to model generalisation. The consistent discrepancy between test and validation performance demonstrates that while the synthetic data helped models optimise for a specific data distribution, it failed to prepare them for the diversity of real-world data. This confirms that evaluation on independent sets is critical to avoid the inflated metrics often seen in literature, a finding that aligns with recent studies calling for more robust evaluation protocols [152, 153].

Our baseline results, after the removal of inter-dataset and intra-dataset and near duplicate URLs, are slightly worse than the near perfect accuracy reported in prior studies. This discrepancy can be attributted to the removal of duplicates, creating a more diverse and generalisable dataset. Despite this, even after the usage of this dataset we were able to achieve results similar to the ones from literature, showing that even after evaluation we can achieve results close to the inflated metrics achieved by them. These inflated metrics align with established findings where performance declines on truly independent evaluation sets confirming that our approach more accurately reflects deployment challenges [152, 153],

It is important to note that the SLM, that was the most powerful experimented technique, showed the worst results. This performance limitation comes from the partial fine-tuning approach, which fails to guarantee phishing URL generation (e.g. https://amazon.co.uk, https://www.google.com).

5.3 Email Aggregator

To evaluate the core generation modules, without the context intervention, we designed a test that combined the outputs of the Email Body Template Generator and the URL Generator. This experiment deliberately excludes the Context KG to assess the baseline capabilities of the generative components on their own.

The methodology involved two steps. First, we used an external model (Gemini 2.5 Pro) to generate 30 common phishing themes, such as “unsuccessful delivery” and “sign in attempt”. These themes served as the initial input for our system. It is important to note that providing

an initial theme is a required step. While a model of this scale could perform open-ended generation, its limited size makes it prone to produce generic or repetitive scenarios without a topic. The usage of a theme helps focusing the generation process and avoids potential pitfalls. Furthermore, allowing human intervention to provide these themes is a practical advantage, as it enables the injection of relevant attack scenarios that a model with a fixed knowledge cut-off date would miss, allowing for more up-to-date and diverse phishing scenarios.

For each theme, the Email Aggregator initiated a multi-stage generation process. It first tasked the Email Body Template Generator with creating a placeholder template and the URL Generator with producing a plausible phishing URL, using the initial theme as their context. These two outputs, along with the original theme, were then supplied to the Gemma 3 12B model. This final model was responsible for assembling all the elements into a complete email, which included creating a fitting sender, which now couldn't be tasked to the Context KG module, and subject line, following the specified language.

The results of this process are presented in Table 5.8, which showcases a selection of the generated emails.

Table 5.8: Selected Examples of Context-Independent Email Generation

Theme	**Generated Output (Sender, Subject, Body Snippet)**
Unsuccessful delivery	**Sender:** notice@postalupdateservices.net **Subject:** Important Notice Regarding Delivery Attempt **Body:** On 10th September 2025 at 11:17 AM We were unable to successfully process a recent postal shipment. Further details can be found by clicking the link below...
Sign in attempt	**Sender:** security@accounts-alert.net **Subject:** Important: Unusual Login Activity Detected - Verify Account **Body:** Dear Mr. Lasse Zikonas, A recent sign-in attempt has triggered a security alert linked to your PayPal account. Please verify it isn't you by clicking the secure link below...
Invoice due	**Sender:** accounts@globalfinancesolutions.com **Subject:** Important: Overdue invoice - Action Required **Body:** Dear Valued Client, You may recall our previous attempts to reach you regarding an outstanding invoice. Please review the details below relating to invoice #INV-20250910-ABC...
Action required	**Sender:** null **Subject:** Error **Body:** Failed to generate or parse email content.

An analysis of the 30 generated emails reveals several findings about the framework's capabilities. Overall, the system is successful at creating plausible, context-free phishing emails. The Gemma 3 12B model shows a strong ability to translate a high-level theme like "Unsuccessful delivery" or "Invoice due" into a complete and coherent message. It consistently crafts a fitting sender, a subject line that conveys urgency or importance, and a body that employs common social engineering tactics. The language is generally fluent, and the generated URLs are thematically appropriate.

A particularly observation is the model's capacity for creative inference, or "hallucination". In the "Sign in attempt" example, the model independently generated a full name (Mr. Lasse Zikonas) and linked the alert to a specific service (PayPal) without any of this information being present in the input theme. While the PayPal service could be derived from the pre-training, the name is a clear signal for hallucination. While technically a hallucination, in this context, it is a beneficial one, as it significantly increases the personalisation and realism of the phishing attack. This goes to show that even without the factual grounding of the KG, a sufficiently large model can draw upon its pre-trained knowledge to invent convincing, context-appropriate details.

However, the experiment also highlights the system's limitations. Out of 30 generation attempts, 2 resulted in failures. One produced an explicit "Failed to generate" error (as shown in Table 5.8), and another resulted in an invalid JSON output due to an encoding error, as documented in its error log. This suggests a failure rate of approximately 6.7%, for the 30 email generated messages.

Chapter 6

Case Study - Porto City Council

Operating within the PERRY architecture, CANDACE is designed to generate realistic phishing and benign emails, which are then used to improve threat detection models. This case study serves the dual purpose of showing the framework's capabilities and, more importantly, proving the value of a context-driven approach to generation.

The framework's development, however, preceded the availability of real-world email data to train the models and test the generation. To test its capabilities under this constraint, we implemented a simulated case study targeting the Porto City Council.

This chapter first details the construction of the KG for the city's operational context, then describes the Email Aggregation process and its core components.

6.1 Knowledge Graph Construction

This section models the KG for the Porto City Council, following the dual-architecture defined in Section 4.4. It is important to note that all information used to populate both graphs was sourced from publicly available data. We will now detail the specific data sources, entities, and relationships that constitute both of these KGs, the External and Internal.

6.1.1 External Knowledge Graph Modelling

The construction of a KG, designed to represent the external operational environments of a city council, began with a data modelling phase.

The initial challenge was to define the scope of the external graph to ensure it captured essential information without becoming too complex. To achieve this balance, the geographic scope was limited to the Porto district. The Porto disctrict will be represented as a `GeographicalLocation`, that encompasses two distinct inter-municipal entities: the *Área Metropolitana do Porto (AMP)* and the *Comunidade Intermunicipal (CIM) do Tâmega e Sousa*.

These two entities collectively represent all municipalities within the Porto district. The AMP includes the councils of *Matosinhos, Maia, Valongo, Vila do Conde, Santo Tirso, Porto, Póvoa de Varzim, Paredes, Gondomar, Vila Nova de Gaia*, and *Trofa*. The *CIM Tâmega e Sousa* comprises the remaining seven municipalities: *Felgueiras, Paços de Ferreira, Lousada, Amarante, Marco de Canaveses, Penafiel*, and *Baião*. Each of these municipalities was modeled as a distinct `GeographicalLocation` node within the KG.

A city council itself is a complex organization, that is represented through its `Governamental Organs`, including the council body (e.g. *Câmara Municipal*), the municipal assembly, the

municipal police, and the civil protection authority. Furthermore, each council needs to operate within a specific legal context. To reflect this, `LegalFrameworks` were also needed to be incorporated into the graph, including the Portuguese Constituion as the nation's fundamental law, the GDPR, Law nº 75/2013 which defines the legal regime for local authorities (*Regime Jurídico das Autarquias Locais*), and the Código do Procedimento Administrativo (CPA) established by Law-Decree nº 4/2015.

The graph also captures the interconnected nature of municipalities. Each city council maintains relationships with its neighbouring district councils and interacts with shared service providers. For councils within the AMP, these include public transport operators such as *Sociedade de Transportes Colectivos do Porto (STCP)* and *Metro do Porto*. Other entities, such as Social Security Institute (*Segurança Social*), the *Instituto do Emprego e Formação Profissional (IEFP)*, serve municipalities across both Intermunicipal regions. The services provided by a city council, such as infrastructures developemnt, social support and development, public health, and public safety, were also defined as nodes.

These services are made available by specific entities and infrastructures, which were modeled as distinct nodes. For public health, seven Local Health Units (LHU) in the Porto district were represented. Each LHU is responsible for one or more hospitals. For instance, the LHU of *São João* manages both the *Hospital de São João* and the *Hospital Nossa Senhora da Conceição*, while the LHU of [Gaia/Espinho] oversees the *Hospital Eduardo Santos Silva* and the *Hospital Distrital de Vila Nova de Gaia*. The other entities, *Matosinhos, Póvoa de Varzim/Vila do Conde, Médio Ave, Tâmega e Sousa*, and *Santo António*, and their respective hospitals were also modelled.

Similarly, Water Management entities were included. This includes companies like *Águas de Matosinhos, EM* and *Águas de Gaia, EM*, as well as specialized operators like *Águas de Valongo - Be Water* for wastewater treatment. Municipalized units, such as the *Serviços Municipalizados de Água e Saneamento* for Lousada, Vila do Conde, and Paredes, were also represented. To complete the service landscape, entities for education (school groupings), waste management, and local fire departments were incorporated.

Once all the entities (nodes) were defined, the next phase involved modelling the relationships (edges) that connect them. These relationships describe the complex interactions and dependencies, using predicates such as `PART_OF`, `LOCATED_IN`, `INTERACTS_WITH`, `MEMBER_OF`, `SERVED_BY`, `HAS_JURISDICTION_OVER`, `GOVERNED_BY`, and `PROVIDES_SERVICE`.

Using the Cypher query language [154], a series of scripts were developed to create these nodes and their relationships, resulting in the complete External KG for the Porto district. Figure 6.1 displays the resulting graph. Although the density of the complete graph makes individual nodes and connections illegible, it is presented to illustrate the overall scale and complexity of the graph, showing the importance of scoping. It is important to note that while some nodes may appear disconnected, this is a visual artifact of the diagram's scale; all nodes are fully connected. The colors distinguish between entity types as follows: blue represents schools (connected to orange school groupings); yellow nodes are fire departments; light blue, legal documents; dark blue, public entities; dark green, services; light green, intermunicipality entities; and brown, water treatment entities.

To provide a clearer view of the graph's structure for a single municipality, a query was executed to isolate the subgraph for the Porto City Council, as shown in Figure 6.2. The resulting graph of the executed query, shown in Figure 6.3, highlights the council's first-degree relationships. In the graph, node colors signify different entity types: orange for school

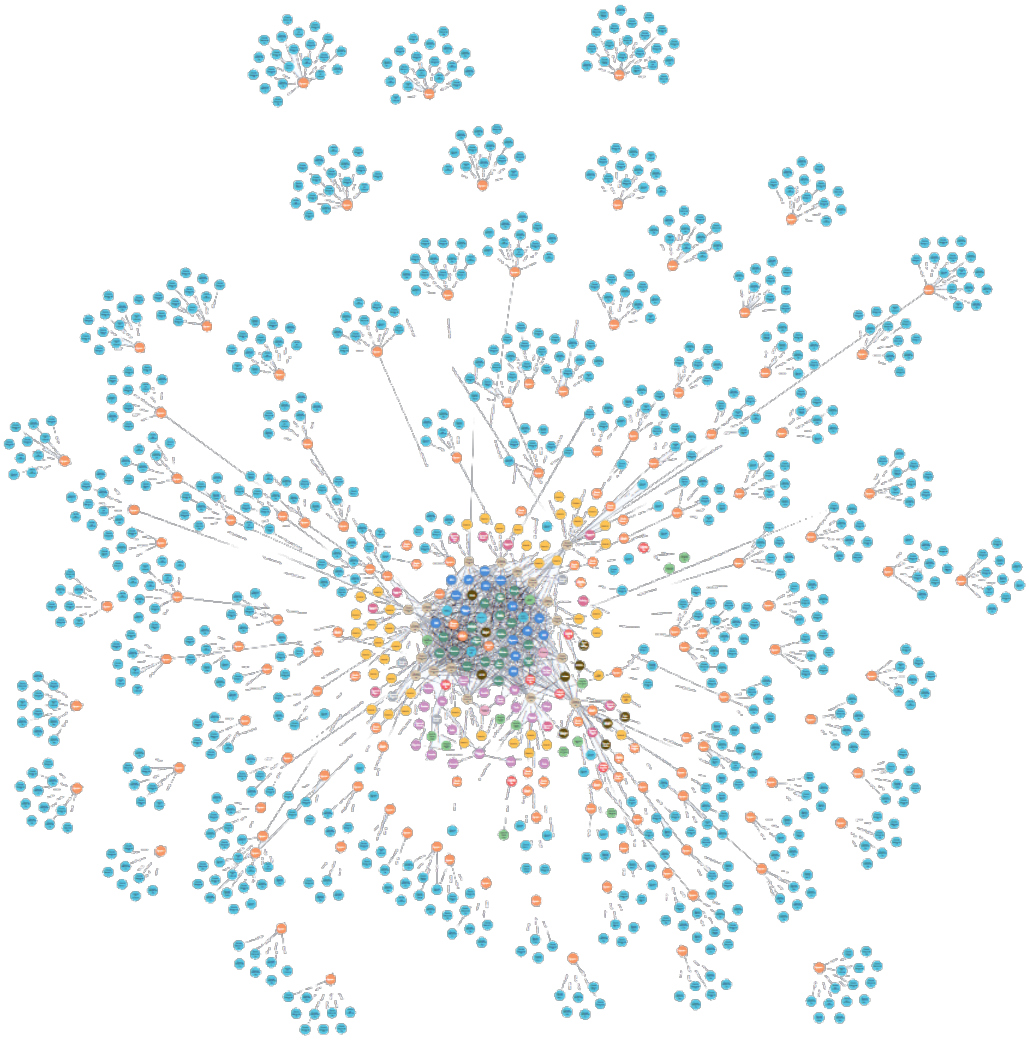

Figure 6.1: External Knowledge Graph for Porto District

groupings, yellow for fire departments, light blue for legal frameworks, red for health units (LHU), light green for the inter-municipality entity (AMP), dark green for services, pink for water management entities, brown for waste management, and other colors for geographical locations and companies. This filtered view demonstrates how the KG effectively maps the immediate operations of a specific city council.

```
MATCH (n:CityCouncil name:"Câmara Municipal de Porto")-[connection]-(connectedNodes)
RETURN n, connection, connectedNodes
```

Figure 6.2: Cypher Query for a Single Municipality

6.1.2 Internal Knowledge Graph Modelling

The construction of the internal KG, designed to represent the internal operations of the Porto City Council, commenced with a detailed data modelling phase. To create a realistic yet manageable model, a set of core entity types was established, representing organizational departments, ongoing projects, critical services, and personnel.

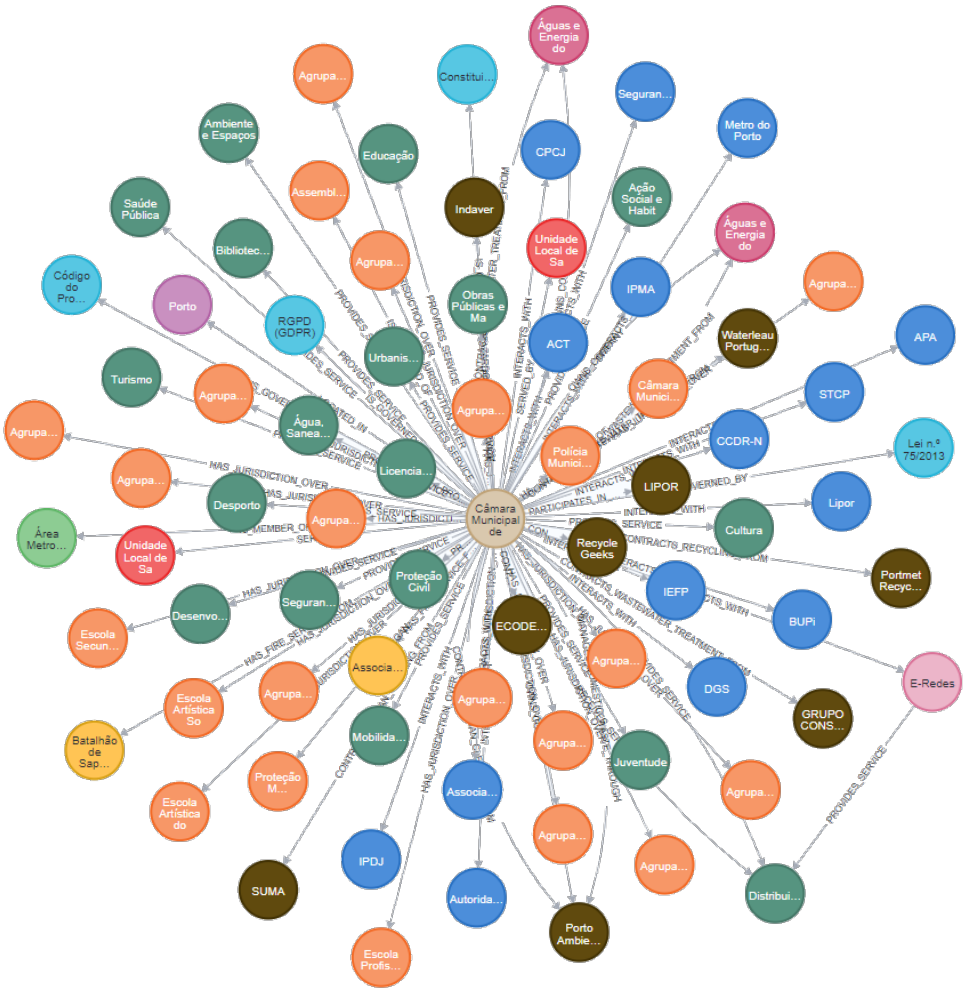

Figure 6.3: External Knowledge Graph for Porto City Council

The council's organizational structure was represented by `Department` nodes, with examples including the *Direção Municipal de Ambiente e Serviços Urbanos*, *Direção Municipal de Finanças*, and *Gabinete da Presidência e Vereação*.

Ongoing strategic and operational initiatives were captured as `Project` nodes. Examples include high-priority undertakings such as the *Expansão da Rede de Metro - Impacto Urbano*, the *Plano de Gestão do Centro Histórico*, and the *Revisão do PDM do Porto*.

A variety of internal and external services crucial for the council's operations were also modeled as `Service` nodes. This encompasses a wide range of suppliers and platforms, including IT solutions like Microsoft 365 and SAP ERP, as well as utility and transport providers such as EDP and STCP.

Finally, to simulate a realistic organizational environment, personnel were modeled as `User` nodes. Each node includes attributes for the user's name, role, and an email address, along with a boolean flag to distinguish between internal and external users. The user base was compiled from publicly available information and supplemented with synthetically generated profiles. For instance, the personnel structure within the *Direção Municipal de Finanças* includes Paulo Fernandes (Diretor Municipal de Finanças) <`paulo.fernandes@cm-porto.pt`>,

Ana Silva (Chefe de Divisão de Contratação Pública) <ana.silva@cm-porto.pt>, "Técnico de Contabilidade 1" (Técnico Superior) <dmf.cont1@cm-porto.pt>, and "Administrativo Tesouraria 1" (Assistente Técnico) <dmf.tes1@cm-porto.pt>. The `User` entity is fundamental, representing not only council employees but also providing a foundation for modelling external actors like citizens or supplier contacts.

The culmination of this modelling process is the internal KG for the Porto City Council, observed in Figure 6.4. This graph illustrates the interconnected network of departments, projects, services, and personnel, providing a comprehensive map of the organization's internal structure and relationships.

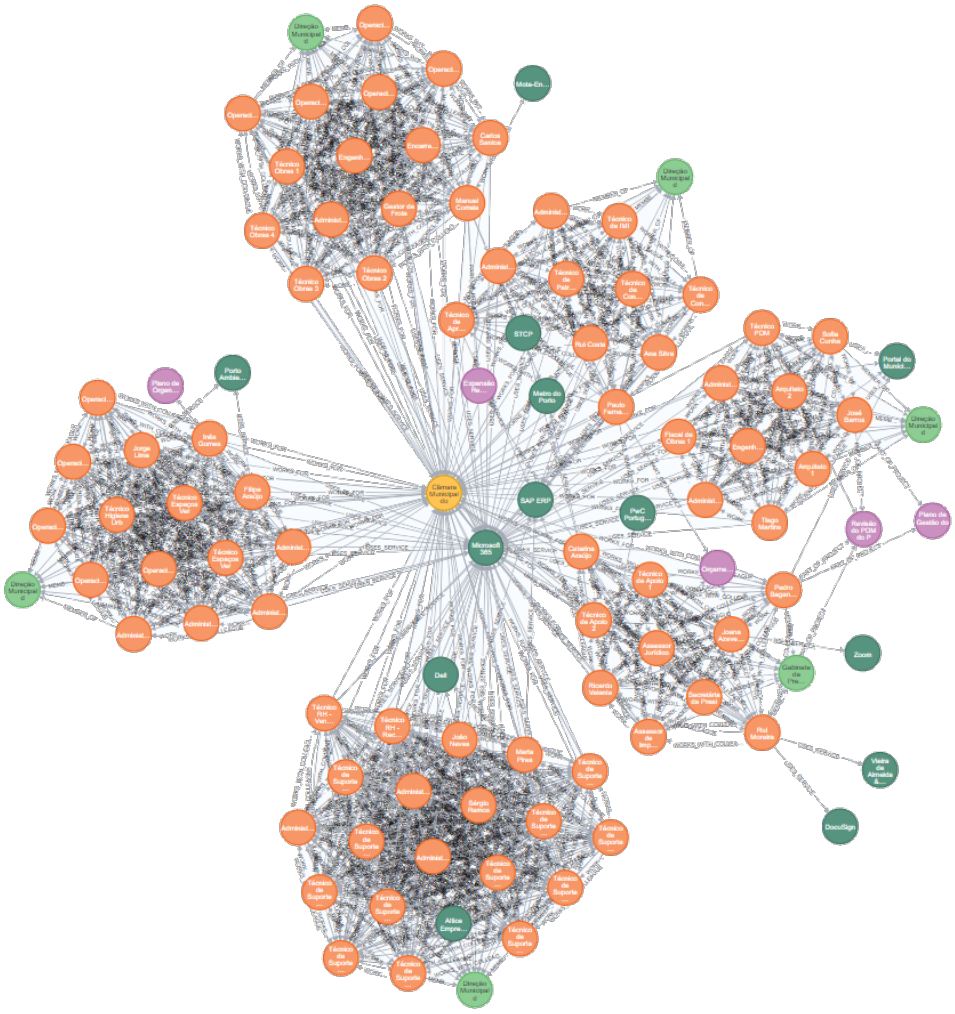

Figure 6.4: Internal Knowledge Graph for the Porto City Council

6.2 Email Aggregation

This section presents the email aggregation process of the case study, implementing the architecture envisioned in Section 4.5.1. Each phase of the architecture is thoroughly discussed to provide a clear exposition of how the constituent components and data sources culminate in the generation of the final, contextually relevant email messages.

6.2.1 Data Collection and Context Improvement Phase

The objective of the initial phase is to gather information to enrich the context of the email to be generated. This context enhancing process relies on querying both Internal and External KG. The aggregated information not only serves to gather more context for developing the email message but also making usage of the Internal KG.

A significant consideration during this phase is the selection of the email's sender. While a sender could be chosen from the organization's internal personnel, simulating phishing attacks that originate from a recognised internal email address was deemed inappropriate for the ethical considerations of this study. For this reason, the sender generation logic is split based on the email's classification as either benign or malicious, that is provided as input for this phase, as it was mentioned in Section 4.4.1.

The data collection and enhancing process begins with querying the Internal KG to obtain a high-level summary of the entire organization's internal structure. This summary encompasses a detailed mapping of departments, active projects, services in use, and personnel. The first query is designed to retrieve all existing departments, currently active projects, and operational services. Following this, a second query groups all users by their respective departments, and a subsequent query identifies any internal users who are not assigned to a department. This information is structured into a clear, human-readable text. In a later stage, the SLM uses this text to make a selection of the sender and recipient. The example of this structured output for the 'Câmara Municipal do Porto' is presented in Figure 6.5.

Subsequently, a similar procedure is applied to the External KG. This query gathers all nodes and connections related to the city council, capturing the name and type of each external entity (e.g., Service Provider, Government Agency, Hospital). This information, which outlines the organization's external ecosystem, is also structured into a readable format, as shown in Figure 6.6.

After acquiring the outputs from both KGs, the SLM (Gemma3 4B Instructor) is used to generate the core context. A problem found during initial experimentation was that the sheer volume of text from the combined internal and external graph data overloaded the model, causing it to default to repetitive and suboptimal personnel selections.

To circumvent this limitation, the context generation task was split into three calls to the SLM. The model is first prompted with the initial email context and instructed to determine the single most relevant department for that context. This strategic reduction of scope ensures that the subsequent generation task operates on a more focused and manageable subset of data, specifically the personnel list of the selected department. Having the personnel list, the recipient is then chosen from it using randomness.

Using the initial theme and the chosen recipient department, the SLM's second task is to determine the most plausible sender. Guided by a set of rules and a classification of being phishing or not, it classifies the sender (e.g., internal, external) and generates a specific persona. In this step, it also proposes a detailed email subject that logically connects the sender, recipient, and theme, using the content of both the Internal and External KGs.

The final step in this phase is the generation of the of the enhanced context. The user prompt, the structure of which is detailed in Figure 6.7 provides all the necessary information for the task: the recipients from the previously identified department, the subject from the second step, a boolean flag (‘is_phishing‘) to guide the sender generation logic, and the subject from the second step to serve as a comprehensive reference.The output of this final

Organizational Structure Summary for 'Câmara Municipal do Porto':

- Departments:
  - Direção Municipal de Ambiente e Serviços Urbanos
  - Direção Municipal de Finanças
  - Direção Municipal de Mobilidade e Obras
  - Direção Municipal de Recursos Humanos e Modernização Administrativa
  - Direção Municipal de Urbanismo
  - Gabinete da Presidência e Vereação

- Active Projects:
  - Expansão da Rede de Metro - Impacto Urbano
  - Orçamento Municipal 2025
  - Plano de Gestão do Centro Histórico
  - Plano de Organização do São João 2024
  - Revisão do PDM do Porto

- Services in Use:
  - Altice Empresas
  - Dell
  - ... (and other services)

- Personnel by Department:
  - Direção Municipal de Ambiente e Serviços Urbanos:
    - Filipe Araújo (Diretor Municipal de Ambiente) <filipe.araujo@cm-porto.pt>
    - Inês Gomes (Chefe de Divisão de Espaços Verdes) <ines.gomes@cm-porto.pt>
    - ... (and other personnel)
  - Direção Municipal de Finanças:
    - Paulo Fernandes (Diretor Municipal de Finanças) <paulo.fernandes@cm-porto.pt>
    - ... (and other personnel)

Figure 6.5: A structured summary of the Internal KG query for 'Câmara Municipal do Porto'. For brevity, the full personnel list is truncated.

step is a structured JSON object containing the selected sender, recipient, and the enhanced context. This object serves as the direct input for the subsequent email generation modules.

6.2.2 URL Generation Phase

The second phase of the architecture is dedicated to the generation of a contextually relevant URLs. This process uses the enhanced context derived from the preceding Data Collection and Context Improvement Phase, as detailed in Section 6.2.1.

The enhanced context serves as the primary input for the SLM, Gemma 3 4B Instructor. The model is guided by a prompt that is dynamically chosen based on whether the email is to be classified as benign or phishing. The prompts, to generate the URLs, provide a specific set of rules and constraints that direct their generation process. For a benign email, the objective is to produce a legitimate and authentic link, whereas for a phishing email, the goal is to create a URL that appears credible but is ultimately deceptive. This ensures that the resulting URL is not only logically consistent with the scenario but also correctly aligned with the email's intended classification.

Facts about the entity 'Câmara Municipal de Porto':

- Contracts hazardous waste from:
  - ECODEAL (Focus: Tratamento de Resíduos Perigosos e Não Perigosos)
  - Indaver (Focus: Gestão de Resíduos Industriais e Perigosos)

- Contracts urban cleaning from:
  - Porto Ambiente (Focus: Gestão de Resíduos Domésticos e Limpeza Urbana)
  - SUMA (Focus: Limpeza Urbana e Gestão de Resíduos Sólidos Urbanos)

- Has jurisdiction over:
  - Agrupamento de Escolas Alexandre Herculano, Porto
  - ... (and other school groupings)

- Interacts with:
  - ACT (Type: State Agency)
  - APA (Type: State Agency)
  - ... (and other agencies)

- Is governed by:
  - Constituição da República Portuguesa
  - RGPD (GDPR)

Figure 6.6: A structured summary of facts retrieved from the External KG related to 'Câmara Municipal do Porto'. The lists have been truncated for brevity.

6.2.3 Template Generation Phase

The third phase of the architecture is responsible for generating the template that establishes the structure and tone of the final email message. Consistent with the URL generation phase, this process is critically dependent on the enhanced context to ensure thematic coherence.

The generation is guided by two principal inputs: the aforementioned enhanced context and the email's classification as either benign or phishing. This classification determines the selection of a corresponding prompt, which provides the model with specific instructions and constraints tailored to the desired outcome.

Furthermore, this phase involves the selection of one of two distinct SLMs: GPT-2 or a Gemma model. The choice of model is determined by the specific requirements of the generation task. If the objective is to produce high-fidelity outputs that adhere strictly to the prompt's instructions, GPT-2 is the preferred choice. Conversely, if the priority is to generate a wider variety of more diverse outputs, the Gemma model is better for that task. The output of this phase is a structured email template, prepared for the final stage of content integration.

The output of this phase is a structured email body template, now semantically grounded by the context and ready for the final stage of aggregation. Table 6.1 presents a selection of these templates generated by the GPT-2 model, illustrating how it translates a formal, structured context into a coherent with placeholders email structure.

You are an AI assistant that writes compelling and specific email scenarios. Your task is to take the pre-determined sender, recipient, and subject, and write a detailed context for an email.

GIVEN DATA:
{{
"sender": json.dumps(sender_details, ensure_ascii=False),
"recipient": json.dumps(chosen_recipient, ensure_ascii=False),
"specific_subject": json.dumps(proposed_subject, ensure_ascii=False),
"is_phishing": json.dumps(is_phishing)
}}

TASK:
Create a final JSON object. The 'enhanced_context' must be a 1-2 sentence scenario that **fully elaborates on the 'Specific Subject'**. It must be a complete and believable message. **DO NOT use placeholders like '[Specify...]' or '[Date]'**. Invent plausible details where necessary.

JSON_OUTPUT_FORMAT:
{{
"recipient": "name": "Recipient Name", "email": "recipient.email@example.com",
"sender": "name": "Sender Name", "email": "sender.email@example.com",
"enhanced_context": "A detailed new context for the email."
}}

YOUR JSON OUTPUT ONLY:

Figure 6.7: The user prompt structure for the email aggregation task, providing the SLM with the necessary data and instructions.

Table 6.1: Examples of Email Body Templates Generated by GPT-2

Theme	Example Generated Template
RE: Queixa sobre estacionamento abusivo	'[GPE]) [NEWLINE] [PERSON]: [NEWLINE] Ok - in what sense do you mean the termination notices? ... can we determine when/if the terminations notice will go down? [NEWLINE] Please let me know. [NEWLINE] Thanks, [NEWLINE] [PERSON]. [END]'
Fatura de serviços de limpeza	'[PAD] suggester ([GPE]) [NEWLINE] Hi folks, [NEWLINE] I found an [ORG] from [PERSON]... The reason I am wondering how it would install with mplayer is that [PERSON] does not allow other media types... Any pointers as to what [ORG] would you recommend? [END]'
Assinatura de Despacho	'[PAD][NEWLINE] Hello [NEWLINE] I don't see anything in the source. When you call my function with any value, is that something else, a macro or library? ... If I have [*ORG*], *is*[DATE] supposed to do the calculation? ... Thanks, [NEWLINE] TheCreator [END]'

6.2.4 Final Email Assembly Phase

The final stage of the architecture is the email assembly phase, where all previously generated components are integrated to produce a comprehensive and contextually relevant email message. This phase serves as the culmination of the entire generation pipeline, synthesizing the outputs from the preceding stages.

The inputs for this phase include the designated sender and recipient, the enhanced context, the selected email template, the generated URL, and the email's classification label (benign or phishing). To further enrich the final output, the complete External KG is also provided as a reference, allowing the model to incorporate additional details that enhance contextual relevance. The current date is also dynamically inserted into the prompt to improve the realism of any temporal references within the email body.

The generation is performed by a Gemma 3 12B Instruct model, selected for its strong instruction-following capabilities. The process is guided by a final prompt, either benign or phishing, corresponding to the email's label, that structures all the input components. This prompt directs the model to integrate the sender, template, URL, and contextual information into a final, coherent email message. The execution of this phase results in the final email artifact, ready for analysis.

6.2.5 Experimental Case Study Results

To demonstrate the practical application and efficacy of the framework, this section presents concrete examples of the synthetically generated email artifacts. The results are the direct output of the multi-phase pipeline detailed previously, showcasing the system's ability to create both benign and malicious emails grounded in the context of the Porto City Council KGs.

The results are organised into two primary examples: a benign email, designed to simulate routine internal or external correspondence, and a phishing email, crafted to mimic a targeted, context-aware attack. For clarity, the end-to-end generation process for each example is documented in Table 6.2 and Table 6.3, respectively.

These examples show that the framework creates emails that are grounded in their context. The benign email's proposal is specific to the city's parks, and the phishing attempt uses the real-world problem of overflowing bins to build a convincing, urgent request. This is a direct contrast to the emails in the previous Chapter, in Table 5.8, where it can be seen that the incorporation of knowledge, further improves the credibility and realism of the email messages. Additionally it also decreases the hallucination, which didn't happen in the context-aware emails.

The tables also highlight the system's flexibility with languages. You can see that the "Enhanced Context" is generated in English, while the final email is written in Portuguese. This is a deliberate design choice. The context generation and the final email writing are separate stages. This separation means the core reasoning can happen in one language, and the final output can be adapted to another. It shows the framework is not locked into a single language and could be implemented for other languages, as well as the modules of it.

Table 6.2: Example of a Generated Benign Email

Component	Generated Content
Initial Context	Email related to Solução
Sender	Inês Gomes <ines.gomes@cm-porto.pt>
Recipient	Filipe Araújo (Diretor Municipal de Ambiente) <filipe.araujo@cm-porto.pt>
Enhanced Context	Following recent discussions with representatives from the 'Espaços Verdes' association regarding community engagement and sustainable waste management within Porto, I am writing to propose a collaborative project focused on establishing designated composting zones strategically located throughout our city's public parks. Specifically, we envision developing three pilot zones - one each in Parque da Cidade, Ribeira Promenade, and Jardins do Comércio - utilizing readily available materials and employing volunteer support to educate park visitors about effective composting practices, ultimately contributing to a greener and more environmentally conscious urban landscape for Porto.
Generated URL	https://www.espacosverdes.pt/compostagem-porto
Final Email Body	Caro Filipe, Espero que este email o encontre bem. Na sequência das recentes discussões com representantes da associação 'Espaços Verdes' relativamente ao envolvimento comunitário e à gestão sustentável de resíduos no Porto, proponho um projeto colaborativo focado na criação de zonas designadas para compostagem estrategicamente localizadas nos nossos parques públicos. A nossa visão é desenvolver três zonas piloto - uma em cada um dos seguintes locais: Parque da Cidade, Passeio Marítimo da Ribeira e Jardins do Comércio - utilizando materiais facilmente acessíveis e contando com apoio voluntário para educar os visitantes sobre práticas eficazes de compostagem. Isto contribuiria significativamente para um panorama urbano mais verde e consciente ambientalmente para o Porto. Poderá encontrar informações adicionais sobre esta iniciativa no seguinte link:https://www.espacosverdes.pt/compostagem-porto Estou disponível para discutir esta proposta com maior detalhe sempre que lhe for conveniente. Com os melhores cumprimentos, Inês Gomes ines.gomes@cm-porto.pt

Table 6.3: Example of a Generated Phishing Email

Component	Generated Content
Initial Context	Email related to A sua encomenda foi devolvida
Sender	Porto Ambiente <portoambiente@cm-porto.pt>
Recipient	Administrativo Ambiente 1 <dmasu.adm1@cm-porto.pt>
Enhanced Context	Following numerous complaints regarding overflowing bins in the city center, the Porto Ambiente team has received a formal request from various residents and local businesses concerning the frequency of collections and overall cleanliness of public areas. Specifically, concerns have been raised about the accumulation of waste near Avenida dos Clérigos due to insufficient collection schedules. We need to address this immediately to maintain public satisfaction and uphold our environmental commitments.
Generated URL	https://portoambiente-updates.com/v2/login/reconfirm-email
Final Email Body	Prezado(a) Administrario Ambiente 1, Porto Ambiente - Diretoria Municipal de Ambiente e Serviços Urbanos informamos que observámos uma atividade incomum associada à sua conta. A recentemente identificámo-nos um problema recorrente com tentativas de acesso não autorizado às contas relacionadas com gestão de resíduos municipais. Esta comunicação visa assegurar a segurança e a integridade dos dados sensíveis relacionados com a administração ambiental do município. Precisamos da sua colaboração imediata para confirmar algumas informações. Por favor, siga o processo de reconfirmação urgente através deste link:https://portoambiente-updates.com/v2/login/reconfirm-email Solicitamos que conclua este procedimento até ao dia 18 de julho de 2025 para evitar potenciais interrupções nos seus serviços na Câmara Municipal do Porto. Este pedido é crítico para manter a conformidade legal junto do RGPD (GDPR) e outros regulamentos aplicáveis. Estamos cientes das preocupações levantadas sobre os depósitos insalubres próximos da Avenida dos Clérigos e trabalharemos proativamente juntos para melhorar nossos níveis de limpeza durante toda esta recuperação de sistema. Mantenha atento aos procedimentos regulares por parte da Empresa Porto Ambiente! Estamos seguros desta solução! Atenciosamente, Porto Ambiente E-mail: portoambiente@cm-porto.pt

Chapter 7

Conclusion

This thesis addressed the challenge of generating high-quality, synthetic email data that is both realistic and contextually relevant, while also adhering to modern data protection standards. The primary objective was to design, implement, and evaluate a system capable of generating emails. This chapter sums up the main conclusions of this work by revisiting the accomplished objectives, before offering a critical reflection on the study's limitations and outlining promising directions for future research.

7.1 Accomplished Objectives

The main objective of this thesis was the generation of synthetic email messages, while also making sure these were contextually relevant. All the established objectives were successfully accomplished, resulting in a system capable of generating synthetic contextually relevant email messages. This solution resulted in many research contributions, with the results for each objective as follows:

- **OB1:** A literature review of the state-of-the-art, presented in Chapter 2 established the theoretical basis for this research. The survey on various architectures, including GANs and PLMs, confirmed that PLMs represent the most suitable approach for this work. The analysis highlighted their superior ability to generate long-form, coherent text and maintain contextual relevance.
- **OB2:** The critical features that differentiate phishing from benign emails were identified through the literature review presented in Chapter 2. The analysis of prior research confirmed that key indicators of phishing include deceptive URL structures, the use of specific psychological cues like urgency and authority, and the impersonation of legitimate entities. This knowledge was then used during the prompt engineering phase in Chapter 4, where these features were explicitly codified as rules within the prompts for the URL Generator and Email Aggregator.
- **OB3:** A modular, generative framework (CANDACE) capable of producing high-quality, context-aware synthetic emails was designed and implemented, as detailed in Chapter 4. This framework includes three novel components: a Email Body Template Generator, a URL Generator, and a KG-informed Context Improvement module that provides factual grounding for the entire process. The innovation of this work was the successful integration of contextual information into the generation process, resulting in synthetic data that is not merely plausible but is semantically grounded and relevant to its domain.

- **OB4:** The quality of the generated content was evaluated in Chapter 5. A significant contribution of this research was the evaluation of how synthetic data influence the performance of ML and DL models. This investigation, which resulted in a peer-reviewed article, examined the domain of URLs generation. It revealed a notable weakness in standard data augmentation practices, demonstrating that models trained on synthetic data may not generalise effectively in real-world scenarios without careful validation against data from different distributions. In addition the analysis in Section 5.1.2 demonstrated a clear trade-off between the fidelity of the GPT-2 model and the superior creativity and novelty of the Gemma 3 model for email body generation, which also resulted in another peer-reviewed article.
- **OB5:** The practical applicability and overall effectiveness of the integrated framework were validated in a simulated real-world scenario, presented as a Case Study in Chapter 6. This case study successfully showed the system's capacity to generate coherent and contextually appropriate email messages for a specific domain (a city council's communication network), proving its utility as a tool for creating high-fidelity synthetic, contextually relevant data.

7.2 Limitation and Future Work

While the research achieved its primary objectives, certain limitations should be acknowledged as they provide context for the results and identify key points for further investigation.

The principal limitation of this study lies in the dataset used for development and testing. The email messages were not specific to the intended application domain. This limited the ability to fully show the system's specialised capabilities. Although the current dataset was good to prove the system's core functionality, a domain-specific corpus would have provided a more rigorous and contextually relevant evaluation of its performance.

The absence of a real-world implementation and evaluation was also a limitation. The system was developed and tested in a controlled, simulated environment. Its performance, scalability, and usability under the dynamic conditions of a live operational setting have not yet been validated. The complexities of a real-world scenario, including user interactions and integration with existing infrastructures, could present challenges not yet encountered during this research.

A next big step is to integrate and evaluate the system using a dataset of email messages directly relevant to the target domain. This would not only allow for a more accurate assessment of the system's effectiveness but also significantly improve the quality and contextual relevance of the generated email messages. By training and testing on domain-specific data, the system could learn better the nuanced language, topics, and conventions of the domain, leading to more valuable outputs.

Furthermore, an objective for future research should be the deployment and testing of the system in a real-world environment. The implementation of this system in the PERRY architecture could be an important way to validate and evaluate its performance. This would allow for the gathering of authentic user feedback and performance metrics, which are essential for identifying areas for refinement and ensuring the system's successful adoption in its intended operational context.

7.3 Final Remarks

This thesis presented the research and development work that led to the creation of an adversarial agent system for the generation of synthetic contextually relevant email messages. This was aligned with the project VESTA at GECAD, both in the field of phishing detection, using synthetic email messages to improve their detection in a specific domain.

Overall, this was a interesting challenge, that made me put to use both my AI knowledge acquired during my master's degree, as well as software engineering acquired during my bachelor's degree. It also allowed me to improve my knowledge in AI, more specific the PLMs field, and the ethical and regulations that need to be put into practice and comply with it, to keep the systems always in check and secure.

Bibliography

[1] *Phishing Landscape 2024*. https://interisle.net/insights/phishing-landscape-2024-an-annual-study-of-the-scope-and-distribution-of-phishing. Accessed: 2024-12-04. July 2024.

[2] *ENISA Threat Landscape 2024*. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024. Accessed: 2024-12-04. Sept. 2024.

[3] *Cyber security breaches survey 2025*. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2025/cyber-security-breaches-survey-2025. Accessed: 2025-07-28. 2025.

[4] *Cost of a Data Breach Report 2024*. https://www.ibm.com/reports/data-breach. Accessed: 2024-12-04. 2024.

[5] *Phishing Activity Trends Reports 1st Quarter*. https://docs.apwg.org/reports/apwg_trends_report_q1_2025.pdf. Accessed: 2025-07-28. 2025.

[6] Lizhen Tang and Qusay H. Mahmoud. "A Survey of Machine Learning-Based Solutions for Phishing Website Detection". In: *Machine Learning and Knowledge Extraction* 3.3 (2021), pp. 672–694. issn: 2504-4990. doi: 10.3390/make3030034. url: https://www.mdpi.com/2504-4990/3/3/34.

[7] Neel Dholakia and Pragati Agrawal. "Review on Phishing Attack Detection Techniques". In: *ASIAN JOURNAL OF CONVERGENCE IN TECHNOLOGY* 6 (Aug. 2020), pp. 41–47. doi: 10.33130/AJCT.2020v06i02.008.

[8] Abeer Alhuzali et al. "In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets". In: *Applied Sciences* 15.6 (2025), p. 3396. issn: 2076-3417. doi: 10.3390/app15063396.

[9] Najwa Altwaijry et al. "Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models". In: *Sensors* 24.7 (2024), p. 2077. issn: 1424-8220. doi: 10.3390/s24072077.

[10] Fatima Zahra Qachfar, Rakesh M. Verma, and Arjun Mukherjee. "Leveraging Synthetic Data and PU Learning For Phishing Email Detection". In: *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*. CODASPY '22. Baltimore, MD, USA: Association for Computing Machinery, 2022, pp. 29–40. isbn: 9781450392204. doi: 10.1145/3508398.3511524. url: https://doi.org/10.1145/3508398.3511524.

[11] Ana Bezerra et al. "A case study on phishing detection with a machine learning net". In: *International Journal of Data Science and Analytics* (June 2024). issn: 2364-4168. doi: 10.1007/s41060-024-00579-w. url: https://doi.org/10.1007/s41060-024-00579-w.

[12] Connor Shorten, Taghi M. Khoshgoftaar, and Borko Furht. "Text Data Augmentation for Deep Learning". In: *Journal of Big Data* 8.1 (July 2021), p. 101. issn: 2196-1115. doi: 10.1186/s40537-021-00492-0. url: https://doi.org/10.1186/s40537-021-00492-0.

[13] Rodrigo Gutiérrez Benítez et al. "Guide for the application of the data augmentation approach on sets of texts in Spanish for sentiment and emotion analysis". In: *PLOS ONE* 19.9 (Sept. 2024), pp. 1–24. doi: 10.1371/journal.pone.0310707. url: https://doi.org/10.1371/journal.pone.0310707.

[14] Gustavo H. de Rosa and João P. Papa. "A survey on text generation using generative adversarial networks". In: *Pattern Recognition* 119 (2021), p. 108098. issn: 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108098. url: https://www.sciencedirect.com/science/article/pii/S0031320321002855.

[15] Mandeep Goyal and Qusay H. Mahmoud. "A Systematic Review of Synthetic Data Generation Techniques Using Generative AI". In: *Electronics* 13.17 (2024). issn: 2079-9292. doi: 10.3390/electronics13173509. url: https://www.mdpi.com/2079-9292/13/17/3509.

[16] Hossein Shirazi et al. "Adversarial Autoencoder Data Synthesis for Enhancing Machine Learning-Based Phishing Detection Algorithms". In: *IEEE Transactions on Services Computing* 16.4 (2023), pp. 2411–2422. doi: 10.1109/TSC.2023.3234806.

[17] Lei Huang et al. "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions". In: *ACM Transactions on Information Systems* (Nov. 2024). issn: 1558-2868. doi: 10.1145/3703155. url: http://dx.doi.org/10.1145/3703155.

[18] Hao Chen et al. *On the Diversity of Synthetic Data and its Impact on Training Large Language Models*. 2024. arXiv: 2410.15226 [cs.CL]. url: https://arxiv.org/abs/2410.15226.

[19] Ken Peffers et al. "The Design Science Research Process: A Model for Producing and Presenting Information Systems Research". In: *1st International Conference, DESRIST 2006 Proceedings*. Claremont Graduate University. 2006, pp. 83–106.

[20] Simon Brown. *The C4 model for visualising software architecture: Context, Containers, Components, and Code*. https://c4model.com/. Accessed: 2025-08-29. 2021.

[21] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a Standard Process Model for Data Mining". In: 2000. url: https://api.semanticscholar.org/CorpusID:1211505.

[22] Francisco Cardoso, Eva Maia, and Isabel Praça. "Email Augmentation: A Comparison of Fine-tuned Small Language Models". In: *Proceedings of the 15th International Conference on Advanced Computer Information Technologies (ACIT)*. 2025.

[23] Francisco Cardoso, Eva Maia, and Isabel Praça. "Improving Machine Learning Models for URL Phishing Detection using Synthetic Data". In: *Computer Security. ESORICS 2025 International Workshops*. Accepted for publication. 2025.

[24] Matthew J. Page et al. "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews". In: *Systematic Reviews* 10.1 (Mar. 2021), p. 89. issn: 2046-4053. doi: 10.1186/s13643-021-01626-4. url: https://doi.org/10.1186/s13643-021-01626-4.

[25] IEEE Xplore Digital Library. *IEEE Xplore*. Accessed: 2024-12-28. https://ieeexplore.ieee.org. 2024.

[26] ACM Digital Library. *ACM Digital Library*. Accessed: 2024-12-28. https://dl.acm.org. 2024.

[27] Web of Science. *Web of Science*. Accessed: 2024-12-28. https://www.webofscience.com. 2024.

[28] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. "A Survey on Data Augmentation for Text Classification". In: *ACM Computing Surveys* 55 (2021), pp. 1–39. url: https://api.semanticscholar.org/CorpusID:235755489.

[29] Jason Wei and Kai Zou. "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6382–6388. doi: 10.18653/v1/D19-1670. url: https://aclanthology.org/D19-1670/.

[30] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. url: https://arxiv.org/abs/2005.14165.

[31] Bin Guo et al. "Conditional Text Generation for Harmonious Human-Machine Interaction". In: *ACM Trans. Intell. Syst. Technol.* 12.2 (Feb. 2021). issn: 2157-6904. doi: 10.1145/3439816. url: https://doi.org/10.1145/3439816.

[32] Wenhao Yu et al. "A Survey of Knowledge-enhanced Text Generation". In: *ACM Comput. Surv.* 54.11s (Nov. 2022). issn: 0360-0300. doi: 10.1145/3512467. url: https://doi.org/10.1145/3512467.

[33] Albert Gatt and Emiel Krahmer. *Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation*. 2018. arXiv: 1703.09902 [cs.CL]. url: https://arxiv.org/abs/1703.09902.

[34] Ahmet Anıl Müngen, Emre DoĞan, and Mehmet Kaya. "Text generation with diversified source literature review". In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ASONAM '19. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2020, pp. 765–770. isbn: 9781450368681. doi: 10.1145/3341161.3343510. url: https://doi.org/10.1145/3341161.3343510.

[35] Bing Li et al. "Advances and challenges in artificial intelligence text generation". In: *Frontiers of Information Technology & Electronic Engineering* 25.1 (Jan. 2024), pp. 64–83. issn: 2095-9230. doi: 10.1631/FITEE.2300410. url: https://doi.org/10.1631/FITEE.2300410.

[36] Qiuyun Zhang et al. "AI-Powered Text Generation for Harmonious Human-Machine Interaction: Current State and Future Directions". In: *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 2019, pp. 859–864. doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00176.

[37] Chenhe Dong et al. "A Survey of Natural Language Generation". In: *ACM Comput. Surv.* 55.8 (Dec. 2022). issn: 0360-0300. doi: 10.1145/3554727. url: https://doi.org/10.1145/3554727.

[38] Baljap Singh et al. "Exploring the Effectiveness of Various Deep Learning Techniques for Text Generation in Natural Language Processing". In: *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*. 2023, pp. 70–75. doi: 10.1109/ICAICCIT60255.2023.10466068.

[39] Abhishek Kumar Pandey and Sanjiban Sekhar Roy. "Natural Language Generation Using Sequential Models: A Survey". In: *Neural Processing Letters* 55.6 (Dec. 2023), pp. 7709–7742. issn: 1573-773X. doi: 10.1007/s11063-023-11281-6. url: https://doi.org/10.1007/s11063-023-11281-6.

[40] Soumaya Loukili, Abdelhadi Fennan, and Lotfi Elaachak. "Applications of Text Generation in Digital Marketing: a review". In: *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*. NISS '23. Larache, Morocco: Association for Computing Machinery, 2023. isbn: 9798400700194. doi: 10.1145/3607720.3608451. url: https://doi.org/10.1145/3607720.3608451.

[41] Rupali Goyal, Parteek Kumar, and V. P. Singh. "A Systematic survey on automated text generation tools and techniques: application, evaluation, and challenges". In: *Multimedia Tools and Applications* 82.28 (Nov. 2023), pp. 43089–43144. issn: 1573-7721. doi: 10.1007/s11042-023-15224-0. url: https://doi.org/10.1007/s11042-023-15224-0.

[42] Sixun Ouyang and Aonghus Lawlor. "Improving Explainable Recommendations by Deep Review-Based Explanations". In: *IEEE Access* 9 (2021), pp. 67444–67455. doi: 10.1109/ACCESS.2021.3076146.

[43] Shen Gao et al. "Meaningful Answer Generation of E-Commerce Question-Answering". In: *ACM Trans. Inf. Syst.* 39.2 (Feb. 2021). issn: 1046-8188. doi: 10.1145/3432689. url: https://doi.org/10.1145/3432689.

[44] Chandan Singh. *Alice in Wonderland Gutenberg*. https://www.kaggle.com/datasets/chandan2495/alice-in-wonderland-gutenbergproject/metadata. Accessed: 2025-08-19. 2017.

[45] Anthony Browne. *Hansel and Gretel*. London: Julia MacRae Books, 1981.

[46] Noureen Fatima et al. "A Systematic Literature Review on Text Generation Using Deep Neural Network Models". In: *IEEE Access* 10 (2022), pp. 53490–53503. doi: 10.1109/ACCESS.2022.3174108.

[47] Junyi Li et al. "Pre-Trained Language Models for Text Generation: A Survey". In: *ACM Comput. Surv.* 56.9 (Apr. 2024). issn: 0360-0300. doi: 10.1145/3649449. url: https://doi.org/10.1145/3649449.

[48] HanQi Jin et al. "Recent advances of neural text generation: Core tasks, datasets, models and challenges". In: *Science China Technological Sciences* 63.10 (Oct. 2020), pp. 1990–2010. issn: 1869-1900. doi: 10.1007/s11431-020-1622-y. url: https://doi.org/10.1007/s11431-020-1622-y.

[49] Xueying Zhang et al. "DSGPT: Domain-Specific Generative Pre-Training of Transformers for Text Generation in E-commerce Title and Review Summarization". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 2146–2150. isbn: 9781450380379. doi: 10.1145/3404835.3463037. url: https://doi.org/10.1145/3404835.3463037.

[50] Li Yuan et al. "Hierarchical template transformer for fine-grained sentiment controllable generation". In: *Inf. Process. Manag.* 59.5 (2022), p. 103048. doi: 10.1016/j.ipm.2022.103048. url: https://doi.org/10.1016/j.ipm.2022.103048.

[51] Yupian Lin et al. "A Survey on Neural Data-to-Text Generation". In: *IEEE Transactions on Knowledge and Data Engineering* 36.4 (2024), pp. 1431–1449. doi: 10.1109/TKDE.2023.3304385.

[52] Hanqing Zhang et al. "A Survey of Controllable Text Generation Using Transformer-based Pre-trained Language Models". In: *ACM Comput. Surv.* 56.3 (Oct. 2023). issn: 0360-0300. doi: 10.1145/3617680. url: https://doi.org/10.1145/3617680.

[53] Touseef Iqbal and Shaima Qureshi. "The survey: Text generation models in deep learning". In: *Journal of King Saud University - Computer and Information Sciences* 34.6, Part A (2022), pp. 2515–2528. issn: 1319-1578. doi: https://doi.org/10.

1016/j.jksuci.2020.04.001. url: https://www.sciencedirect.com/science/article/pii/S1319157820303360.

[54] Yingli Shen and Xiaobing Zhao. "Reinforcement Learning in Natural Language Processing: A Survey". In: *Proceedings of the 2023 6th International Conference on Machine Learning and Natural Language Processing*. MLNLP '23. Sanya, China: Association for Computing Machinery, 2024, pp. 84–90. isbn: 9798400709241. doi: 10.1145/3639479.3639496. url: https://doi.org/10.1145/3639479.3639496.

[55] Lantao Yu et al. "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (2017). doi: 10.1609/aaai.v31i1.10804. url: https://ojs.aaai.org/index.php/AAAI/article/view/10804.

[56] Rashi Agarwal, Himanshu Agarwal, and Senam Pandey. "Unveiling the Depths: A Comprehensive Analysis of Natural Language Processing and Generative Adversarial Neural Networks for Text Generation Models in Deep Learning". In: *2023 1st International Conference on Circuits, Power and Intelligent Systems (CCPIS)*. 2023, pp. 1–6. doi: 10.1109/CCPIS59145.2023.10291966.

[57] Heng Wang, Zengchang Qin, and Tao Wan. "Text Generation Based on Generative Adversarial Nets with Latent Variables". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Dinh Phung et al. Cham: Springer International Publishing, 2018, pp. 92–103. isbn: 978-3-319-93037-4.

[58] Tong Che et al. *Maximum-Likelihood Augmented Discrete Generative Adversarial Networks*. 2017. arXiv: 1702.07983 [cs.AI]. url: https://arxiv.org/abs/1702.07983.

[59] Mansoureh Motahari Nezhad and Mohammadreza Kangavari. "Personalized Persuasive Text Generation". In: *2024 10th International Conference on Artificial Intelligence and Robotics (QICAR)*. 2024, pp. 261–267. doi: 10.1109/QICAR61538.2024.10496651.

[60] K. Chitra, G. Kavitha, and P. Latchoumy. "Penalty based Sentimental Text Generation Framework using Generative Adversarial Networks". In: *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*. 2022, pp. 1147–1152. doi: 10.1109/ICACRS55517.2022.10029135.

[61] K. Wang and X. Wan. "Automatic generation of sentimental texts via mixture adversarial networks". In: *Artificial Intelligence* 275 (2019), pp. 540–558. issn: 0004-3702. doi: https://doi.org/10.1016/j.artint.2019.07.003. url: https://www.sciencedirect.com/science/article/pii/S0004370218306088.

[62] Kevin Lin et al. *Adversarial Ranking for Language Generation*. 2018. arXiv: 1705.11001 [cs.CL]. url: https://arxiv.org/abs/1705.11001.

[63] Jiaxian Guo et al. *Long Text Generation via Adversarial Training with Leaked Information*. 2017. arXiv: 1709.08624 [cs.CL]. url: https://arxiv.org/abs/1709.08624.

[64] Samuel R. Bowman et al. *Generating Sentences from a Continuous Space*. 2016. arXiv: 1511.06349 [cs.LG]. url: https://arxiv.org/abs/1511.06349.

[65] Wenlin Wang et al. "Topic-Guided Variational Auto-Encoder for Text Generation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 166–177. doi: 10.18653/v1/N19-1015. url: https://aclanthology.org/N19-1015.

[66] Yifan Li et al. *Diffusion Models for Non-autoregressive Text Generation: A Survey*. 2023. arXiv: 2303.06574 [cs.CL]. url: https://arxiv.org/abs/2303.06574.

[67] Q. Yi et al. "Diffusion models in text generation: a survey". In: *PeerJ Computer Science* 10 (2024), e1905. doi: 10.7717/peerj-cs.1905.

[68] Shansan Gong et al. *DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models*. 2023. arXiv: 2210.08933 [cs.CL]. url: https://arxiv.org/abs/2210.08933.

[69] Hongyi Yuan et al. *SeqDiffuSeq: Text Diffusion with Encoder-Decoder Transformers*. 2023. arXiv: 2212.10325 [cs.CL]. url: https://arxiv.org/abs/2212.10325.

[70] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. "DiffuSum: Generation Enhanced Extractive Summarization with Diffusion". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 13089–13100. doi: 10.18653/v1/2023.findings-acl.828. url: https://aclanthology.org/2023.findings-acl.828.

[71] Xiang Li et al. "Diffusion-LM Improves Controllable Text Generation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 4328–4343. url: https://proceedings.neurips.cc/paper_files/paper/2022/file/1be5bc25d50895ee656b8c2d9eb89d6a-Paper-Conference.pdf.

[72] Zhengfu He et al. "DiffusionBERT: Improving Generative Masked Language Models with Diffusion Models". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 4521–4534. doi: 10.18653/v1/2023.acl-long.248. url: https://aclanthology.org/2023.acl-long.248.

[73] Robin Strudel et al. *Self-conditioned Embedding Diffusion for Text Generation*. 2022. arXiv: 2211.04236 [cs.CL]. url: https://arxiv.org/abs/2211.04236.

[74] Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. "SSD-LM: Semi-autoregressive Simplex-based Diffusion Language Model for Text Generation and Modular Control". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 11575–11596. doi: 10.18653/v1/2023.acl-long.647. url: https://aclanthology.org/2023.acl-long.647.

[75] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423. url: https://aclanthology.org/N19-1423.

[76] Alec Radford et al. "Improving Language Understanding by Generative Pre-Training". In: *OpenAI Blog*. 2018. url: https://openai.com/blog/language-unsupervised/.

[77] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Proceedings of the 34th International Conference on Machine Learning*. 2020. url: http://proceedings.mlr.press/v119/raffel20a.html.

[78] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics. 2020. url: https://www.aclweb.org/anthology/2020.acl-main.703/.

[79] Zeqiu Wu et al. *Automatic Document Sketching: Generating Drafts from Analogous Texts*. 2021. arXiv: 2106.07192 [cs.CL]. url: https://arxiv.org/abs/2106.07192.

[80] Linan Zhu et al. "Fine-Grained Sentiment-Controlled Text Generation Approach Based on Pre-Trained Language Model". In: *Applied Sciences* 13.1 (2023). issn: 2076-3417. doi: 10.3390/app13010264. url: https://www.mdpi.com/2076-3417/13/1/264.

[81] Timon Felske, Sebastian Bader, and Thomas Kirste. "Automatic Generation of Personalised and Context-Dependent Textual Interventions During Neuro-rehabilitation". In: *KI - Künstliche Intelligenz* 36.2 (Sept. 2022), pp. 189–193. issn: 1610-1987. doi: 10.1007/s13218-022-00765-7. url: https://doi.org/10.1007/s13218-022-00765-7.

[82] Hongting Zheng Zheyi Chen Liuchang Xu. "Evolution and Prospects of Foundation Models: From Large Language Models to Large Multimodal Models". In: *Computers, Materials & Continua* 80.2 (2024), pp. 1753–1808. issn: 1546-2226. doi: 10.32604/cmc.2024.052618. url: http://www.techscience.com/cmc/v80n2/57626.

[83] Liangjing Shao et al. "Artificial intelligence generated content (AIGC) in medicine: A narrative review". In: *Mathematical Biosciences and Engineering* 21.1 (2024), pp. 1672–1711. issn: 1551-0018. doi: 10.3934/mbe.2024073. url: https://www.aimspress.com/article/doi/10.3934/mbe.2024073.

[84] Rong Xiang et al. "Cantonese natural language processing in the transformers era: a survey and current challenges". In: *Language Resources and Evaluation* (June 2024). issn: 1574-0218. doi: 10.1007/s10579-024-09744-w. url: https://doi.org/10.1007/s10579-024-09744-w.

[85] Cuiyun Gao et al. "Automating App Review Response Generation Based on Contextual Knowledge". In: *ACM Trans. Softw. Eng. Methodol.* 31.1 (Oct. 2021). issn: 1049-331X. doi: 10.1145/3464969. url: https://doi.org/10.1145/3464969.

[86] Nina Dethlefs and Heriberto Cuayáhuitl. "Hierarchical Reinforcement Learning for Adaptive Text Generation". In: *Proceedings of the 6th International Natural Language Generation Conference*. Ed. by John Kelleher, Brian Mac Namee, and Ielka van der Sluis. Association for Computational Linguistics, July 2010. url: https://aclanthology.org/W10-4204.

[87] Zhan Shi et al. "Toward Diverse Text Generation with Inverse Reinforcement Learning". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 4361–4367. doi: 10.24963/ijcai.2018/606. url: https://doi.org/10.24963/ijcai.2018/606.

[88] Pierre Dognin et al. "ReGen: Reinforcement Learning for Text and Knowledge Base Generation using Pretrained Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1084–1099. doi: 10.18653/v1/2021.emnlp-main.83. url: https://aclanthology.org/2021.emnlp-main.83.

[89] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July

2002, pp. 311–318. doi: 10.3115/1073083.1073135. url: https://aclanthology.org/P02-1040/.

[90] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. url: https://aclanthology.org/W04-1013/.

[91] Frederick Jelinek et al. "Perplexity—a measure of the difficulty of speech recognition tasks". In: *Journal of the Acoustical Society of America* 62 (1977). url: https://api.semanticscholar.org/CorpusID:121680873.

[92] 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*. Ed. by Jade Goldstein et al. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. url: https://aclanthology.org/W05-0909/.

[93] Tianyi Zhang et al. *BERTScore: Evaluating Text Generation with BERT*. 2020. arXiv: 1904.09675 [cs.CL]. url: https://arxiv.org/abs/1904.09675.

[94] Jiwei Li et al. "A Diversity-Promoting Objective Function for Neural Conversation Models". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, June 2016, pp. 110–119. doi: 10.18653/v1/N16-1014. url: https://aclanthology.org/N16-1014/.

[95] Ke Wang and Xiaojun Wan. "SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 4446–4452. doi: 10.24963/ijcai.2018/618. url: https://doi.org/10.24963/ijcai.2018/618.

[96] Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. "TESLA: Translation Evaluation of Sentences with Linear-Programming-Based Analysis". In: *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Ed. by Chris Callison-Burch et al. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 354–359. url: https://aclanthology.org/W10-1754/.

[97] Bhuwan Dhingra et al. "Handling divergent reference texts when evaluating table-to-text generation". In: *arXiv preprint arXiv:1906.01081* (2019).

[98] Wei Zhao et al. *MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance*. 2019. arXiv: 1909.02622 [cs.CL]. url: https://arxiv.org/abs/1909.02622.

[99] Trivikram Muralidharan and Nir Nissim. "Improving Malicious Email Detection Through Novel Designated Deep-Learning Architectures Utilizing Entire Email". In: *Neural Networks* 157 (2023), pp. 257–279. issn: 0893-6080. doi: 10.1016/j.neunet.2022.09.002. url: https://www.sciencedirect.com/science/article/pii/S0893608022003367.

[100] Debalina Bera, Obi Ogbanufe, and Dan J. Kim. "Towards a thematic dimensional framework of online fraud: An exploration of fraudulent email attack tactics and intentions". In: *Decision Support Systems* 171 (2023), p. 113977. issn: 0167-9236. doi: https://doi.org/10.1016/j.dss.2023.113977. url: https://www.sciencedirect.com/science/article/pii/S0167923623000520.

[101] Tarini Saka, Kami Vaniea, and Nadin Kökciyan. "Context-Based Clustering to Mitigate Phishing Attacks". In: *Proceedings of the 15th ACM Workshop on Artificial*

Intelligence and Security. AISec'22. Los Angeles, CA, USA: Association for Computing Machinery, 2022, pp. 115–126. isbn: 9781450398800. doi: 10.1145/3560830.3563728. url: https://doi.org/10.1145/3560830.3563728.

[102] Said Salloum et al. "A New English/Arabic Parallel Corpus for Phishing Emails". In: *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 22.7 (July 2023). issn: 2375-4699. doi: 10.1145/3606031. url: https://doi.org/10.1145/3606031.

[103] Panagiotis Bountakas, Konstantinos Koutroumpouchos, and Christos Xenakis. "A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection". In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*. ARES '21. Vienna, Austria: Association for Computing Machinery, 2021. isbn: 9781450390514. doi: 10.1145/3465481.3469205. url: https://doi.org/10.1145/3465481.3469205.

[104] Brij B. Gupta et al. "Advanced BERT and CNN-Based Computational Model for Phishing Detection in Enterprise Systems". In: *Computer Modeling in Engineering & Sciences* 141 (2024). issn: 1526-1506. doi: 10.32604/cmes.2024.056473. url: http://www.techscience.com/CMES/v141n3/58510.

[105] Sijie Zhuo et al. "A Large-Scale Study of Device and Link Presentation in Email Phishing Susceptibility". In: *Proceedings of the 35th Australian Computer-Human Interaction Conference*. OzCHI '23. Wellington, New Zealand: Association for Computing Machinery, 2024, pp. 78–85. isbn: 9798400717079. doi: 10.1145/3638380.3638434. url: https://doi.org/10.1145/3638380.3638434.

[106] Sijie Zhuo et al. "Eyes on the Phish(er): Towards Understanding Users' Email Processing Pattern and Mental Models in Phishing Detection". In: *Proceedings of the 2024 European Symposium on Usable Security*. EuroUSEC '24. Association for Computing Machinery, 2024, pp. 15–29. isbn: 9798400717963. doi: 10.1145/3688459.3688465. url: https://doi.org/10.1145/3688459.3688465.

[107] Matt Dixon et al. "Holding Your Hand on the Danger Button: Observing User Phish Detection Strategies Across Mobile and Desktop". In: *Proc. ACM Hum.-Comput. Interact.* 6.MHCI (Sept. 2022). doi: 10.1145/3546730. url: https://doi.org/10.1145/3546730.

[108] Ayoub Alsarhan et al. "Enhancing Phishing URL Detection: A Comparative Study of Machine Learning Algorithms". In: *Proceedings of the 2023 Asia Conference on Artificial Intelligence, Machine Learning and Robotics*. AIMLR '23. Bangkok, Thailand: Association for Computing Machinery, 2023. isbn: 9798400708312. doi: 10.1145/3625343.3625348. url: https://doi.org/10.1145/3625343.3625348.

[109] Canadian Institute for Cybersecurity. *URL dataset (ISCX-URL2016)*. https://www.unb.ca/cic/datasets/url-2016.html. Accessed: 2025-08-20. 2025.

[110] ebubekirbbr. *pdd: input directory*. https://github.com/ebubekirbbr/pdd/tree/master/input. Accessed: 2025-08-20. 2025.

[111] Santosh Kumar Birthriya, Priyanka Ahlawat, and Ankit Kumar Jain. "Enhanced Phishing Website Detection Using Dual-Layer CNN and GRU with Attention Mechanism and Lexical NLP Features". In: *SN Computer Science* 5.7 (Oct. 2024), p. 929. issn: 2661-8907. doi: 10.1007/s42979-024-03282-6. url: https://doi.org/10.1007/s42979-024-03282-6.

[112] Mariya Shmalko et al. "Profiler: Distributed Model to Detect Phishing". In: *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. 2022, pp. 1336–1337. doi: 10.1109/ICDCS54860.2022.00152.

[113] Quan Hong Nguyen et al. "Utilizing Large Language Models with Human Feedback Integration for Generating Dedicated Warning for Phishing Emails". In: *Proceedings of*

the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems. SecTL '24. Singapore, Singapore: Association for Computing Machinery, 2024, pp. 35–46. isbn: 9798400706912. doi: 10.1145/3665451.3665531. url: https://doi.org/10.1145/3665451.3665531.

[114] Jingyu Tong and Xuefang Zhang. "Intellivoting: Detection Method of Phishing Emails Based on Hybrid Features and Soft Voting Mechanism". In: *Proceedings of the 2023 4th International Conference on Big Data Economy and Information Management*. BDEIM '23. Zhengzhou, China: Association for Computing Machinery, 2024, pp. 673–677. isbn: 9798400716669. doi: 10.1145/3659211.3659327. url: https://doi.org/10.1145/3659211.3659327.

[115] D Ferlin Deva Shahila et al. "AI Based Phishing Discrement for Immense E-Maildata". In: *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)* 1 (2024), pp. 270–277. url: https://api.semanticscholar.org/CorpusID:272722269.

[116] Shahrzad Sayyafzadeh et al. "Securing Against Deception: Exploring Phishing Emails Through ChatGPT and Sentiment Analysis". In: *2024 IEEE/ACIS 22nd International Conference on Software Engineering Research, Management and Applications (SERA)*. 2024, pp. 159–165. doi: 10.1109/SERA61261.2024.10685564.

[117] Springer Nature. *SpringerLink*. Accessed: 2024-12-28. Retrieved from https://link.springer.com. 2024.

[118] Parisa Mehdi Gholampour and Rakesh M. Verma. "Adversarial Robustness of Phishing Email Detection Models". In: *Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics*. IWSPA '23. Charlotte, NC, USA: Association for Computing Machinery, 2023, pp. 67–76. isbn: 9798400700996. doi: 10.1145/3579987.3586567. url: https://doi.org/10.1145/3579987.3586567.

[119] Abbas Jabr Saleh Albahadili, Ayhan Akbas, and Javad Rahebi. "Detection of phishing URLs with deep learning based on GAN-CNN-LSTM network and swarm intelligence algorithms". In: *Signal, Image and Video Processing* 18.6 (Aug. 2024), pp. 4979–4995. issn: 1863-1711. doi: 10.1007/s11760-024-03204-2. url: https://doi.org/10.1007/s11760-024-03204-2.

[120] Rania Zaimi, Mohamed Hafidi, and Mahnane Lamia. "A deep learning mechanism to detect phishing URLs using the permutation importance method and SMOTE-Tomek link". In: *The Journal of Supercomputing* 80.12 (Aug. 2024), pp. 17159–17191. issn: 1573-0484. doi: 10.1007/s11227-024-06124-7. url: https://doi.org/10.1007/s11227-024-06124-7.

[121] Hayk Ghalechyan et al. "Phishing URL detection with neural networks: an empirical study". In: *Scientific Reports* 14.1 (Oct. 2024), p. 25134. issn: 2045-2322. doi: 10.1038/s41598-024-74725-6. url: https://doi.org/10.1038/s41598-024-74725-6.

[122] Sakshi Mahendru and Tejul Pandit. "SecureNet: A Comparative Study of DeBERTa and Large Language Models for Phishing Detection". In: *2024 IEEE 7th International Conference on Big Data and Artificial Intelligence (BDAI)*. 2024, pp. 160–169. doi: 10.1109/BDAI62182.2024.10692765.

[123] Casey Hanks et al. "Domain Independent Deception Detection: Feature Sets, LIWC Efficacy, and Synthetic Data Challenges". In: *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics*. IWSPA '24. Porto, Portugal: Association for Computing Machinery, 2024, pp. 59–68. isbn: 9798400705564. doi: 10.1145/3643651.3659895. url: https://doi.org/10.1145/3643651.3659895.

[124] Leigh Metcalf and Jonathan M. Spring. "The Ecosystem of Detection and Blocklisting of Domain Generation". In: *Digital Threats* 2.3 (June 2021). doi: 10.1145/3423951. url: https://doi.org/10.1145/3423951.

[125] The European Parliament and the Council of the European Union. "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)". In: *Official Journal of the European Union*. L 119 (May 2016). Regulation 2016/679, pp. 1–88. url: http://data.europa.eu/eli/reg/2016/679/oj.

[126] The European Parliament and the Council of the European Union. "Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act)". In: *Official Journal of the European Union*. L (July 2024). Regulation 2024/1689, pp. 1–144. url: http://data.europa.eu/eli/reg/2024/1689/oj.

[127] Alec Radford et al. *Language Models are Unsupervised Multitask Learners*. Tech. rep. Accessed: 2025-04-29. OpenAI, 2019.

[128] Gemma Team. *Gemma 3 Technical Report*. 2025. arXiv: 2503.19786 [cs.CL].

[129] William W. Cohen. *Enron Email Dataset*. https://www.cs.cmu.edu/∼./enron/. Accessed: 2025-05-12. 2015.

[130] Jose Nazario. *Nazario Phishing Corpus*. https://monkey.org/∼jose/phishing. Accessed: 2025-05-12. 2005.

[131] Dragomir Radev. *CLAIR Collection of Fraud Email*. ACL Data and Code Repository, ADCR2008T001. http://aclweb.org/aclwiki. 2008.

[132] Gordon V. Cormack. *TREC 2005 Public Spam Corpus*. https://plg.uwaterloo.ca/∼gvcormac/treccorpus/about.html. Accessed: 2025-05-12.

[133] Gordon V. Cormack and Thomas R. Lynam. *TREC 2006 Public Spam Corpus*. https://plg.uwaterloo.ca/∼gvcormac/treccorpus06/about.html. Accessed: 2025-05-12.

[134] Gordon V. Cormack and Thomas R. Lynam. *TREC 2007 Public Spam Corpus*. https://plg.uwaterloo.ca/∼gvcormac/treccorpus07/about.html. Accessed: 2025-05-12.

[135] Gordon V. Cormack. *CEAS 2008 Live Spam Challenge Corpus*. https://plg.uwaterloo.ca/∼gvcormac/ceascorpus/. Accessed: 2025-05-12. 2008.

[136] Michael Han Daniel Han and Unsloth team. *Unsloth*. 2023. url: http://github.com/unslothai/unsloth.

[137] Arvind Prasad and Shalini Chandra. *PhiUSIIL Phishing URL (Website)*. UCI Machine Learning Repository. 2024. doi: https://doi.org/10.1016/j.cose.2023.103545.

[138] Abdelhakim Hannousse and Salima Yahiouche. *Web page phishing detection*. Version V3. 2021. doi: 10.17632/c2gw7fy2j4.3. url: https://doi.org/10.17632/c2gw7fy2j4.3.

[139] Harisudhan411. *Phishing and Legitimate URLs*. Accessed: 2025-05-12. 2023. url: https://www.kaggle.com/datasets/harisudhan411/phishing-and-legitimate-urls.

[140] Sid321axn. *Malicious URLs Dataset*. Accessed: 2025-05-12. 2023. url: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset/data.

[141] Ahmed AlEroud and George Karabatis. "Bypassing Detection of URL-based Phishing Attacks Using Generative Adversarial Deep Neural Networks". In: *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*. IWSPA '20. New Orleans, LA, USA: Association for Computing Machinery, 2020, pp. 53–60. isbn: 9781450371155. doi: 10.1145/3375708.3380315. url: https://doi.org/10.1145/3375708.3380315.

[142] Yukun Li et al. "A stacking model using URL and HTML features for phishing webpage detection". In: *Future Generation Computer Systems* 94 (2019), pp. 27–39. issn: 0167-739X. doi: 10.1016/j.future.2018.11.004. url: https://www.sciencedirect.com/science/article/pii/S0167739X1830503X.

[143] Shaoxiong Ji et al. "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (2022), pp. 494–514. doi: 10.1109/TNNLS.2021.3070843.

[144] Neo4j. *Neo4j Graph Data Platform*. Web Page. Accessed: 2025-04-29. 2025. url: https://neo4j.com.

[145] Brij Gupta et al. "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment". In: *Computer Communications* 175 (July 2021), pp. 47–57. doi: 10.1016/j.comcom.2021.04.023.

[146] Adarsh Mandadi et al. "Phishing Website Detection Using Machine Learning". In: *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*. 2022, pp. 1–4. doi: 10.1109/I2CT54291.2022.9824801.

[147] Sajjad Jalil, Muhammad Usman, and Alvis Fong. "Highly accurate phishing URL detection based on machine learning". In: *Journal of Ambient Intelligence and Humanized Computing* 14.7 (July 2023), pp. 9233–9251.

[148] Saleem Raja Abdul Samad et al. "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection". In: *Electronics* 12.7 (2023). issn: 2079-9292. doi: 10.3390/electronics12071642. url: https://www.mdpi.com/2079-9292/12/7/1642.

[149] Er. Kritika. "A comprehensive literature review on phishing URL detection using deep learning techniques". In: *Journal of Cyber Security Technology* 0.0 (2024), pp. 1–29. doi: 10.1080/23742917.2024.2378552. eprint: https://doi.org/10.1080/23742917.2024.2378552. url: https://doi.org/10.1080/23742917.2024.2378552.

[150] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[151] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[152] Fariza Rashid et al. "Phishing URL detection generalisation using Unsupervised Domain Adaptation". In: *Computer Networks* 245 (2024), p. 110398. issn: 1389-1286. doi: https://doi.org/10.1016/j.comnet.2024.110398. url: https://www.sciencedirect.com/science/article/pii/S1389128624002305.

[153] Pedro Afonso et al. "Rethinking Phishing Detection: How Dataset Quality Affects Model Generalization". In: *Proceedings of the 15th International Conference on Advanced Computer Information Technologies (ACIT)*. 2025.

[154] Nadime Francis et al. "Cypher: An Evolving Query Language for Property Graphs". In: *Proceedings of the 2018 International Conference on Management of Data*. SIGMOD

'18. Houston, TX, USA: Association for Computing Machinery, 2018, pp. 1433–1445. isbn: 9781450347037. doi: 10.1145/3183713.3190657. url: https://doi.org/10.1145/3183713.3190657.