The Indonesian Journal of Computer Science
www.ijcs.net
Volume 15, Issue 2, April 2026
https://doi.org/10.33022/ijcs.v15i2.4982

Machine Learning Techniques to Detect Social Engineering Attacks in Text-Based Communications: A Systematic Review

Thomas Maseko[1], Michael Moeti[2], Karabo Mokganya[3]
karabomokganya@gmail.com[3]
[1,2] Faculty of Computer Science, Tshwane University of Technology, Pretoria, South Africa
[3] Department of Public Affairs, Faculty of Humanities, Tshwane University of Technology, Pretoria, South Africa

Article Information

Received : 25 Aug 2025
Revised : 15 Feb 2026
Accepted : 31 Mar 2026

Keywords

Social engineering, phishing detection, NLP, machine learning

Abstract

In the digital age, social engineering-type attacks have posed a significant threat in the cybersecurity space. The act of Social Engineering attacks is the act of leveraging psychological manipulation to deceive individuals into divulging confidential information or performing harmful actions. Detecting Social Engineering is challenging due to the emergence of artificial intelligence and contextual subtleties. Therefore, to improve on cybersecurity posture, this systematic review explores (ML) machine learning techniques designed to identify social engineering attacks in text-based communications, by analysing the performance, methodologies, and limitations of ML techniques. Machine learning techniques in the detection of social engineering attacks have progressed from simple lexical classifiers to sophisticated deep contextual models. The engine of this paper is an empirical systematic review that identifies trends, gaps, and strengths in current literature. Paucity speaking on literature, PRISMA is used for systematically surveying articles aligned with the detection of social engineering attacks using ML techniques. The ability of machine learning techniques to effectively identify various forms of SE attacks and adapt to emerging threats makes ML a great tool in combating Social Engineering attacks. As the volume of digital communication persistently grows at an unprecedented rate, so does the potential of criminals to exploit these channels. The researchers concluded with a recommendation for a comprehensive survey of ML techniques for detecting social engineering attacks in text-based communications

A. Introduction

With the proliferation of internet usage and digital communication platforms, social engineering attacks have become a challenge and a serious threat in the digital age [18]. Social engineering attacks are a type of cyber threat that leverages psychological manipulation to deceive individuals into divulging sensitive information or performing certain actions that compromise security (Aung & Yamana, 2019). The majority of individuals are not aware of social engineering techniques, and do not comprehend the extent of damage SE attacks may cause (Bezuidenhout et al. 2010). Some individuals may not have the understanding that the information at their disposal may be of no particular value, nor utilised for any malicious acts. Phishing attacks accounted for over 76% of cybersecurity type of attacks in 2023 according to Data Breach Investigations Report 2023. Unlike network structures, web applications, desktop and mobile applications, technical protection measures are mostly ineffective against social engineering attacks (Ozkaya, 2018). Earlier counter efforts relied on rule-based filters and blacklisting, which had limited capabilities to adapt to novel attack patterns. The increased adoption of (AI) artificial intelligence and (ML) machine learning, attackers leverage AI capabilities to craft highly personalised textual-based scams [40]. Machine learning models are utilized to analyse datasets from communication platforms, corporate websites, and social media sites to tailor deceiving communications i.e. AI tools that mimic writing styles and replicate organisational jargon to manipulate the victims. According to the literature, the classification of social engineering attacks can be put in three categories, namely human based social engineering attacks, computer based social engineering attacks, Social based attacks (He et al. 2022). Human based attacks are when the attacker interacts with the target to obtain or hoax the individual into disclosing confidential information (He et al. 2022). This method does not require complex programs or skills but relies on human social intervention. Secondly, computer-based social engineering attacks rely on the attackers using devices to access confidential information, such as credit card information and passwords (He et al. 2022). Thirdly, social-based attacks are reliant on the use of psychological and social means to develop a relationship with the victim (He et al. 2022). Text-based social engineering attacks have seen an escalation due to the ability of artificial intelligence to generate context aware messages at scale and natural sounding [23]. This paper discusses the social engineering typologies, ML techniques used in the detection of social engineering attacks, and modelling challenges in the detection of SE attacks in text-based communications.

Table 1. Types of social engineering attacks (He et al. 2022)

Types of social engineering attacks	
Phishing	Impersonation of trusted individuals or entities with the objective of tricking the targeted individual into providing sensitive information.
Vishing	The usage of voice messaging or phone calls to deceive individuals into providing personal information.

Pretexting	Scammers create fabricated scenarios to deceive targeted individuals into providing unauthorised access to systems or providing sensitive information.
Baiting	Victims are lured with rewards or attractive offers.
Tailgating	An unauthorised person/s gaining digital or physical access to restricted areas by following an authorised individual.
Quid pro quo	A benefit or service is offered in exchange for access or sensitive information.
Scareware	The use of fear or deception to manipulate victims into providing personal information or downloading unwanted software.
Dumpster diving	Searching through a targeted individual's digital trash or physical discarded trash.

Social engineering attacks are increasingly conducted through text based communication due to anonymity. Traditional security mechanisms often fall short in detecting social engineering attacks, due to the reliance on human vigilance and predefined rules [36]. The application of machine learning techniques in this domain can be presented with a couple of challenges, including the risk of false positives, large and diverse datasets, and the variability of attack strategies [36]. This paper undertakes a systematic review of machine learning techniques for detecting social engineering attacks in text based communication platforms, assessing the adaptive modeling strategies to enhance robustness and operational readiness.

B. Research Method

This systematic review is written to answer the following research questions:

- RQ1: What are the most commonly used machine learning techniques for text-based social engineering attacks?
- RQ2: How can machine learning techniques effectively detect social engineering attacks in text-based communications?
- RQ3: What are the primary challenges and limitations in applying machine learning techniques in the detection of social engineering attacks?

To answer the above-mentioned research questions, literature published between 2018 and 2025 is rigorously examined. PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses) a reporting guideline for systematic reviewing, is utilized. PRIMSA provides a four-phase flow diagram and a 27-item checklist for reporting essential aspects of literature review [24]. The main objective of PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses) is to ensure that systematic reviews are reported in a manner that allows readers to assess the validity of the findings and imitate the methodology if needed [24]. The four phases of PRISMA include identification, an activity of comprehensively searching bibliographic databases; screening, which involves removing duplicate records; eligibility, which involves retrieval and full text review of all literature passing the screening phase; and inclusion, which is the final selection of all the literature meeting all eligibility criteria[24]. The four

phases aid in ensuring transparency and reproducibility in how literature is identified, screened, and included in the final analysis:

B.1. Inclusion and Exclusion

To ensure a rigorous and reproducible systematic review this literature follows an inclusion and criteria. Studies were selected according to the following predefined criteria.

Table 2. Inclusion Criteria

Inclusion	Details
Language	Literature written in English
Focus area	Research and literature specifically addressing the application of ML techniques in the detection of social engineering attacks
Study type	Published conference papers, peer-reviewed journal articles, and accredited technical reports
Publication date	Studies published between 2018 to 2025
Attack types	Phishing attacks, vishing transcripts, smashing, human-targeted deception delivered via text, fraudulent chat conversations, and social media-based social engineering attacks
Methodology	Literature employing empirical methods, inclusive of quasi-experimental and case studies
Evaluation metrics	Research includes performance metrics

These criteria ensure that the systematic review focuses on machine learning driven text analysis for social engineering attacks aligned with the research objective, and balances depth and breadth.

Table 3. Exclusion Criteria

Exclusion	Details
Language	Literature not written in English
Irrelevant focus	Research and literature that does not address social engineering attacks and or does not apply ML techniques to detect social engineering attacks
Insufficient data	Literature lacking comprehensive information on evaluation metrics and datasets
Theoretical papers	Literature that is purely theoretical with any empirical validation
Duplicate Publications	multiple publications reporting identical methods and results without substantive extension

B.2. Search Process:

The search process that was conducted is accordance with PRISMA guidelines to identify literature studies on ML techniques for detecting social engineering attacks. The process followed four stages namely: database selection, search string formulation, record retrieval and screening.

B.3. Study Selection and Quality Assessment:

The quality appraisal tool utilized for this paper is PROBAST (Prediction model Risk of Bias Assessment Tool) with a customized quality appraisal checklist tailored for the evaluation of ML literature focused on the detection of SE attacks.

The customized quality appraisal checklist comprises of study design and objectives, data quality and preprocessing, feature engineering, model development, evaluation and validation, bias, fairness and robustness. In addition, a structured data extraction template was used. The data extraction template included fields such as study title, authors, year, data source, ML techniques, features used, evaluation metrics, dataset size, validation method, performance results, bias/ fairness analysis, and reproducibility.

B.4. Search Method

Research databases where utilized to mutually capture computer science and interdisciplinary security literature. Literature searches were conducted on five major research databases namely; IEEE Xplore, ACM Digital Library, Scopus, web of Science, Google Scholar using search terminologies to identify research literature focusing on machine learning techniques to detect engineering attacks in text based communications. Only literature dating from 2018 onwards was considered. Articles are scored according to the exclusion and inclusion criteria.

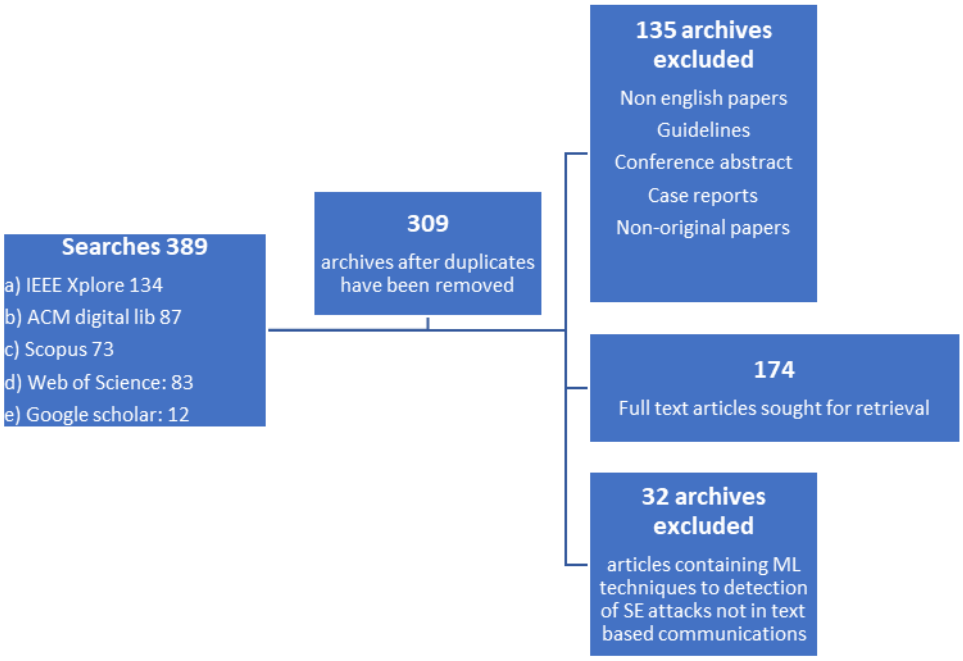

Figure 1. Exclusion Diagram

C. Machine Learning Techniques for The Detection of Social Engineering

Social engineering attacks mostly manifest through text based channels such as chat platforms, emails, social media platforms and SMS, with phishing attacks being the most prominent form. Literature has demonstrated the feasibility of ML techniques in the detection of social engineering attacks [38]. Earlier machine learning approaches applied traditional classifiers such as **Naive Bayes (NB), Random Forests (RF)**, **and Support Vector Machines (SVM)** to detect phishing attacks [13]. These models often utilise handcrafted features, including structural email properties, n-gram frequencies, lexical cues and sender metadata. Common indicators used to distinguish malicious messages were the presence of keywords,

string length and URL anomalies [13]. Email has remained the most common vector for social engineering attacks. Various ML techniques have been deployed in the detection of social engineering attacks, including:

- **Natural language processing (NLP)**: utilises models to analyse linguistic patterns in order to identify anomalies that may indicate social engineering attacks [9]. Natural language processing algorithms analyse urgency and tone in textual communication to identify manipulative tactics common in phishing attacks (Lansley et al. 2019). Further, Natural language processing models identify deviations from normal communication styles such as grammatical errors, unusual syntax and vocabulary shifts [9]. Named entity recognition (NER) extracts and classifies entities to identify suspicious elements such as mismatched sender addresses [9]. NER aids with the detection of impersonation attempts by cross-checking entities with known legitimate sources. Transformer based models like GPT and BERT can understand and evaluate the context and nuances to distinguish legitimate requests, making them effective in detecting subtle cues of deception [9].
- **Logistic Regression**: one of the most interpretable ML classifiers used in the detection of social engineering attack sin text based communications. LR predicts the probability of a binary outcome based on a weighted linear combination of input features [33]. Logistic regression has been extensively utilised in the detection of phishing emails by analysing features such as embedded URLs, linguistic urgency, spoofed sender addresses [2]. In the detection of social media scams LR algorithms analyse direct messages and posts to identify scams by analysing sentiment polarity, and user behavior. LR is effective for binary classification tasks, particularly in distinguishing between legitimate and malicious messages, by analysing features extracted from textual communication such as presence of suspicious keywords, word frequency and metadata, this is called Term Frequency-Inverse Document Frequency [33]. The effectiveness of detection relies on linguistic features such as N-gram analysis. N-gram analysis detects the sequence of characters or words common in a text scam [2]. The model's performance is evaluated using metrics such precision, accuracy F1-score and recall to ensure effectiveness [2].
- **Naïve Bayes**: NB classifiers work by calculating the possibility that a given message belongs to a particular class based on the frequency of phrases and words, this offers fast training and prediction suitable for high volume environments [7]. The effectiveness of Naïve Bayes in the domain of social engineering detection is reliant on feature selection such as structural features, lexical features and metadata features
- **Decision trees**: DT classifies text as malicious through a hierarchical decision making process by analysing linguistic patterns, metadata and structural features [38]. Decision trees creates a tree like model of decisions based on the features of the text, each decision point is represented by a node in the tree [7]. DT offers a balanced approach in the detection of text based social engineering attacks, espousing interpretability with effective classification [38].

- **Gradient boosting (XGBoost)**: gradient boosting combines high predictive accuracy with computational efficiency. It improves upon traditional decision trees by sequential tree building, regularization, weighted feature handling, and parallel processing [27]. XGBoost combines multiple learners, typically decision trees in the creation of strong predictive models. Each tree is constructed chronologically, with each new built tree correcting the errors of the previous ones [27]. XGBoost can handle imbalanced data by using techniques like oversampling/ undersampling and adjusting the class weights, this helps with social engineering datasets that contain more legitimate messages than malicious ones [27].
- **Random Forest**: ensemble classifiers that construct multitude decision trees utilizing bootstrap samples and feature subsampling, then aggregate their predictions via majority voting. This approach enhances generalization, reduces overfitting, and naturally handles high-dimensional feature qualities advantageous for detection of social engineering attacks in text-based communication [27].

Table 4. Six Widely-Used Approaches for Detecting Social Engineering Attacks in Text-Based Communications

Comparison of ML approaches				
Technique	**Feature Representation**	**Strengths**	**Weaknesses**	**Best use case**
Natural language processing (NLP)	Tokenization; POS tags; parse trees; static embeddings (Word2Vec, GloVe); contextual embeddings (BERT, RoBERTa)	• Effective for real time detection • Captures syntactic and semantic nuance • Detects subtle cues (e.g. sentiment shifts, discourse structure) • Multilingual and domain-adaptable	• Large pre-trained models require fine-tuning • High memory and inference cost • Prone to domain shift without adaptation	Advanced phishing detection, deepfake text analysis
Logistic Regression	Handcrafted features (TF-IDF n-grams; keyword counts; readability metrics); PCA-reduced embeddings	• Easy and simple to implement • Highly interpretable coefficients • Effective for binary classification • Fast inference • Low memory footprint	• May not capture complex patterns • Linear decision boundary limits expressiveness • Vulnerable to adversarial obfuscation • Requires careful feature engineering	Baseline phishing detection
Naïve Bayes	Bag-of-words or n-gram counts (Multinomial/Bernoulli variants); TF-IDF; chi-square–selected features	• Fast and scalable • Handles high-dimensional sparse data • Performs well with small datasets	• Strong feature-independence assumption, which is often unrealistic • Performance degrades on obfuscated text • Cannot capture term dependencies or sequence information	Real-time email filtering, spam detection
Decision trees	Handcrafted lexical and syntactic features; structural cues (HTML ratio; URL tokens);	• Handles both numerical and categorical data well • Intuitive rule-based decisions	• Prone to overfitting noisy text data • Poor generalization without pruning or ensembling	Explainable fraud detection

	readability scores	• Easy to visualize	• Can be unstable with small variations in data	
Gradient boosting (XGBoost)	Handcrafted and embedding features (TF-IDF; BERT embeddings; metadata); feature-subsampled decision tree	• Handles missing data well • Handles heterogeneous features • Built-in regularization reduces overfitting • Provides feature importance rankings	• Does not model sequential dependencies • Requires careful hyperparameter tuning • Higher inference latency than single trees	Enterprise phishing detection, high-stakes fraud analysis
Random Forest	TF-IDF, embeddings, syntactic and semantic features; bootstrap-aggregated decision trees	• Provides reliable feature importance • Robust to noisy/adversarial features • Handles high-dimensional data without heavy tuning	• Decision consensus can obscure individual tree insights • Limited ability to capture very long-range semantics • Ensemble size increases memory footprint	Large-scale email security, multi-channel threat detection

The systematic review of 32 studies published between 2018 to 2025 on ML techniques for the detection of social engineering attacks in text-based communications identified several convergent themes and common limitations.

Table 5. Core ML Approaches Integrative Comparison

Core ML Approaches Integrative Comparison			
Technique category	**Key methodologies**	**Common strengths**	**Common limitations**
Linguistic Analysis	• Semantic role labeling • Stylometry (syntax/lexical features) • Sentiment/urgency detectors	• High precision (85-92%) for generic phishing Interpretable features	• 78% of studies confirm >90% F1-score decline when tested on adversarial text
Behavioral Modeling	• UEBA (communication baselines) • Graph neural networks (relationship mapping) • Temporal pattern analysis	• Exceptional BEC detection (89-94% precision) Resilient to linguistic obfuscation	• High false positives during crises GDPR compliance challenges
Hybrid Systems	• NLP + metadata fusion • Ensemble models (e.g., RF + BERT) • Reinforcement learning for adaptive thresholds	• Highest overall accuracy (93-97%) Reduces false positives by 40%	• Complex deployment High computational load

D. Model Comparison and Evaluation Metrics

This evaluation metrics studies the six widely used approaches for detecting social engineering attacks namely; natural language processing, logistic regression, Naïve bayes, decision tress and random forest. The analysis focuses on quantitative metrics, computational efficiency, interpretability and practical applicability.

Table 5. Model-Specific Evaluation Summaries

Evaluation Summaries		
Model	**Typical Performance Metrics**	**Comments / Use Case Considerations**
Natural Language Processing (NLP)	• **Accuracy**: 0.90–0.96 • **Precision**: 0.91–0.95 • **Recall**: 0.89–0.94	Highly effective in detecting nuanced language and adversarial obfuscation. Slightly prone to overfitting without

	• **F1-Score**: 0.90–0.94 • **ROC-AUC**: 0.95+	domain adaptation.
Logistic Regression (LR)	• **Accuracy**: 0.85–0.90 • **Precision**: 0.83–0.89 • **Recall**: 0.80–0.87 • **F1-Score**: 0.82–0.88 • **ROC-AUC**: 0.88–0.91	Performs well with engineered features; interpretability is a strong benefit. Struggles with complex language cues.
Naïve Bayes (NB)	• **Accuracy**: 0.86–0.89 • **Precision**: 0.84–0.88 • **Recall**: 0.85–0.89 • **F1-Score**: 0.85–0.88 • **ROC-AUC**: 0.87–0.90	High recall makes NB suitable for broad detection, but the independence assumption limits nuanced pattern capture.
Decision Tree (DT)	• **Accuracy**: 0.84–0.88 • **Recall**: 0.82–0.88 • **F1-Score**: 0.82–0.87 • **ROC-AUC**: 0.85–0.89	Easy to interpret but prone to overfitting. Better suited for exploratory work or when rule transparency is key.
Random Forest (RF)	• **Accuracy**: 0.88–0.93 • **Precision**: 0.89–0.93 • **Recall**: 0.87–0.92 • **F1-Score**: 0.88–0.92 • **ROC-AUC**: 0.94+	High resilience to noise and strong interpretability via feature importance. Less effective on sequential/textual structure
Gradient Boosting (XGBoost)	• **Accuracy**: 0.90–0.94 • **Precision**: 0.91–0.95 • **Recall**: 0.89–0.94 • **F1-Score**: 0.90–0.94 • **ROC-AUC**: 0.96+	Strong performance with heterogeneous features. Balanced precision-recall profile. Ideal for operational deployment

In the domain of social engineering detection, the performance of the model must be evaluated not only by accuracy but by the metrics that reflect the asymmetric costs of misclassification [9]. While simpler models like Naïve bayes and logistic regression offer speed and interpretability, methods such as XGboost and random forest provide superior performance in complex and real world scenarios the best model depends on the data characteristics, operational context and the acceptable trade-off between false positives and false negatives (Schmitt and Flechais. 2024).

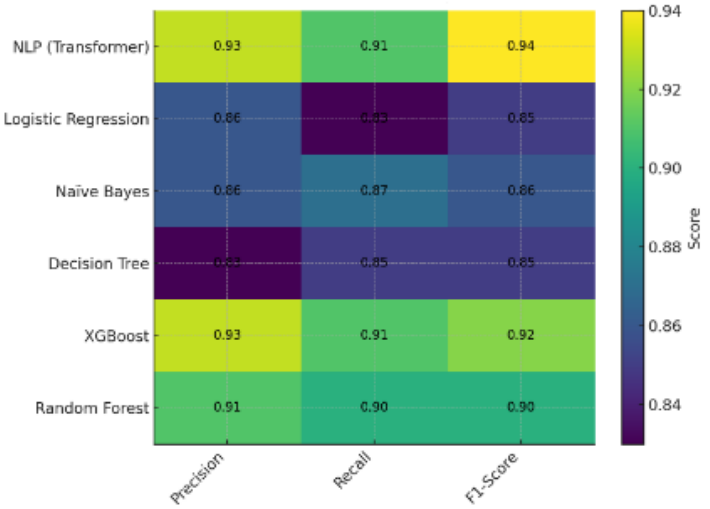

Figure 2. Performance heatmap of ML models for social engineering detection

The heatmap compared key performance metrics precision, recall, and F1-score across six model classes for social engineering detection. This visualisation highlights that **transformer-based NLP** achieves the highest F1-score (0.94), albeit with high precision and recall, **XGBoost** closely follows, balancing strong precision and recall (both ~0.92–0.93), **Traditional models** (Logistic Regression, Naïve Bayes, Decision Tree) provide solid baselines around F1 ≈ 0.85, with Decision Trees slightly lower in precision, **Random Forest** offers a midpoint, with balanced metrics (~0.90) and robustness. the dominance of NLP and deep learning like transformer based models is widely utilised due to their contextual understanding. Phishing datasets are commonly utilised such as PhishTank, Enron and Enron, with fewer addressing social media and SMS. The dominant evaluation metrics for ML techniques used in the detection of social engineering attacks are accuracy, precision, recall and F1-score, fewer metrics evaluate real-time performance, robustness and adversarial resistance. The common gaps identified are that most datasets are synthetic, lacking real-world diversity and complexity. There's a lack of explainable AI approaches tailored to social engineering attacks. Another gap is the lack of human centric defences, because most models operate in isolation, without integration into awareness systems, user training or human in the loop frameworks.

E. Challenges and limitations

The detection of social engineering attacks in text-based communication presents unique challenges that expose limitations across various machine learning approaches.

- **Natural language processing (NLP):** Social engineering messages often exploit ambiguous or context-sensitive language, making it difficult for even advanced NLP systems to distinguish benign from malicious intent (Zhou et al. 2020). NLP models often struggle with detecting implicit intent or subtle linguistic cues used in sophisticated social engineering attacks, such as psychological manipulation, euphemistic language, or disguised authority (Li et al. 2021). Pretrained models like BERT may underperform on phishing-specific corpora unless fine-tuned with domain-relevant data (Li et al. 2021). NLP models are susceptible to **textual adversarial attacks such as** small changes in wording, misspellings or homographs (Raman et al. 2020). Many social engineering attacks rely on external context such as time of day time of day, sender identity, organisational role which NLP systems often ignore unless augmented. **Another challenge of NLP is high computational costs,** meaning that training and deploying transformer-based models demand significant computational resources (Devlin et al. 2019). NLP models are prone to adversarial perturbations meaning attackers can manipulate inputs (misspellings, homographs) to evade detection (Li et al. 2021).
- **Logistic Regression**: Logistic Regression assumes a linear relationship between input features and the outcome variable, which limits its ability to capture complex patterns in phishing messages. It Requires extensive feature engineering, and its performance is highly sensitive to the choice and quality of textual features such as TF-IDF, n-grams. Logistic Regression

struggles with capturing interactions between features such as linguistic tone and metadata (James et al. 2021). In large and high-dimensional text spaces, Logistic Regression may underperform compared to tree-based or deep learning models.

- **Naïve Bayes:** Naïve Bayes assumes that all features are independent given the class label—an assumption rarely true in natural language contexts (Achary and Shelke. 2023). One of the limitations of Naïve Bayes is The independence assumption often leads to misclassification in texts with complex syntactic structures or contextual dependencies. may overemphasise certain terms without understanding contextual nuance, leading to higher false positives or negatives (Achary and Shelke. 2023).
- **Decision Tree:** Decision Trees tend to overfit on training data, particularly in noisy or imbalanced phishing datasets and Small changes in the dataset can drastically alter the tree structure and output [33]. The instability is caused by small changes in the dataset that drastically alter the tree structure and output. In terms of generalisation, Pure decision trees often lack robustness when faced with unseen phishing attack patterns or evolving threats [33]. Unlike probabilistic models DT do not provide reliable confidence scores for predictions.
- **Gradient Boosting (XGBoost):** XGBoost models are highly sensitive to parameter tuning, which can be computationally expensive and require expert knowledge (Raman et al. 2020). In imbalanced datasets, XGBoost may favour frequent patterns, potentially missing novel or rare social engineering tactics. As an ensemble method, the interpretability of XGBoost is limited, complicating its use in regulated environments where explainability is critical. In terms of resource intensive, training and testing on large datasets can be time- and resource-intensive compared to simpler models (Raman et al. 2020).
- **Random Forest:** While more robust than individual trees, Random Forests can become computationally inefficient when a large number of trees or features are involved [2]. Although feature importance can be extracted, the decision-making process of Random Forests is generally opaque. The ensemble nature of Random Forests makes them less suitable for time-sensitive or real-time attack detection scenarios [2]. Like XGBoost, Random Forests can still suffer from performance degradation on minority classes unless balanced training methods are applied.

Across all machine learning models several overarching limitations persist, data scarcity and class imbalance in Social engineering datasets are often imbalanced, with far fewer malicious examples than benign, which affects the generalisability of all models [9]. Attackers continually adapt, altering language and tactics to evade detection. Static models degrade in performance over time unless continuously updated (Schmitt and Flechais. 2024). In terms of Ethical and Legal Constraints, using communication data raises privacy concerns and compliance issues, which restrict the availability of high-quality training data.

Most models lack continuous learning for evolving attacks such as AI generated social engineering attacks. Current attacks apply a multimodal approach meaning text only detection will fail against voice and text hybrid attacks. With

real time deepfake texting, static NLP models won't be effective against current threat trends.

F. Conclusion

The detection of social engineering attacks in textual based communication environments demands a multifaceted approach. Machine learning techniques apply detection of social engineering attacks by automating the identification of deceptive linguistic patterns, behavioral cues anomalies and metadata that distinguish malicious messages. The six most commonly used ML models for SE detection is NLP, logistic regression, Naïve Bayes, decision tree, random forest, and gradient boosting. Social engineering attacks remains a complex and evolving challenge in the field of cybersecurity. Transformer-based NLP models deliver state of the art performance often exceeding F1-scores of 0.92 by capturing contextual and long-range dependencies critical for identifying subtle deception. However, their reliance on extensive compute resources and vulnerability to adversarial perturbations constrain their standalone deployment in latency-sensitive or resource-constrained environments [22]. Among the models evaluated Naïve bayes and logistic regression offer simplicity making them suitable for baseline comparisons and rapid deployment. However, their reliance on linear assumptions and feature independence respectively limits their effectiveness in capturing the nuanced and context rich nature of deceptive language [7]. In practice, a hybrid detection pipeline combining fast, interpretable filters (e.g., Naïve Bayes or shallow Decision Trees) with robust ensemble classifiers and deep NLP analysers could yield resilient in the defence against evolving social engineering tactics. Future research should focus on continual learning to adapt models to emerging attack patterns, domain adaptation for cross-platform robustness, and lightweight attention-based architectures that reconcile semantic depth with operational efficiency. Such integrative strategies will be essential for deploying scalable, explainable, and enduring social engineering detection systems in real-world settings. Fewer studies address deep flake, IA generated attacks and multi-model SEA such as combining text with visual or audio. Also limited study has been done on models testing against adversarially crafted messages. As social engineering tactics continue to evolve in sophistication and subtlety, future research in the detection of these attacks within text-based communication must advance along several critical dimensions.

G. References

[1] Abu-Nimeh, S., Nappa, D., Wang, X. and Nair, S. (2007) 'A comparison of machine learning techniques for phishing detection', Proceedings of the eCrime Researchers Summit, pp. 60–69.

[2] Achary, R. and Shelke, C.J. (2023) 'An Expert System for the Detection and Mitigation of Social Engineering Attacks Using Machine Learning Algorithm', SpringerLink. Available at: https://link.springer.com/content/pdf/10.1007/978-981-19-5443-6_29.pdf

[3] Ahmed, M.M. (2022). Social Engineering Attacks in E-Government System: Detection and Prevention. International Journal of Applied Engineering and Management Letters, pp.100–116.
[4] Alemayehu, N.Y. (2023) 'Social Engineering attack detection model', Master's thesis, St. Mary's University, St Mary University.
[5] Alhogail, A. and Alsabih, A. (2021). Applying Machine Learning and Natural Language Processing to Detect Phishing Email. Computers & Security, p.102414.
[6] Alsariera, Y.A., Adeyemo, V.E., Balogun, A.O. and Alazzawi, A.K. (2020). AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites. IEEE Access, 8, pp.142532–142542.
[7] Alsufyani, A.A. and Alzahrani, S.M. (2021). ' Social Engineering Attack Detection Using Machine Learning: Text Phishing Attack ', Indian Journal of Computer Science and Engineering, 12(3), pp.743–751. doi:https://doi.org/10.21817/indjcse/2021/v12i3/211203298.
[8] Basit, A., Zafar, M., Liu, X., Javed, A.R., Jalil, Z. and Kifayat, K. (2020). A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommunication Systems, 76(1).
[9] Batool, N. and Ali, A. (2024) 'Using NLP for Social Engineering Detection: AI-Based Approaches to Prevent Cyber Manipulation'
[10] Bergholz, A., De Beer, J., Glahn, S. et al. (2010) 'New filtering approaches for phishing email', Journal of Computer Security, 18(1), pp. 7–35.
[11] Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. Big Data & Society, 3(1), pp.1–12.
[12] Catal, C., Giray, G., Tekinerdogan, B., Kumar, S. and Shukla, S. (2022). Applications of deep learning for phishing detection: a systematic literature review. Knowledge and Information Systems, 64(6), pp.1457–1500.
[13] Gupta, C. and Sharma, M. (2021) 'Hybrid detection of phishing attacks using text embeddings and network features', IEEE Transactions on Information Forensics and Security', Vol 16, pp. 3178–3189.
[14] Hadnagy, C. (2018). Social engineering: the science of human hacking. [online] Indianapolis, In Wiley. Available at: https://www.wiley.com/en-us/Social+Engineering%3A+The+Science+of+Human+Hacking%2C+2nd+Edition-p-9781119433385 [Accessed 22 May 2025].
[15] Heartfield, R. and Loukas, G. (2015). A Taxonomy of Attacks and a Survey of Defence Mechanisms for Semantic Social Engineering Attacks. ACM Computing Surveys, 48(3), pp.1–39. doi:https://doi.org/10.1145/2835375.
[16] Heartfield, R. and Loukas, G. (2018) 'Detecting semantic social engineering attacks with the weakest link: Implementation and empirical evaluation of a human-as-a-security-sensor framework', Computing and Information Systems, vol. 76, pp. 101–127, ISSN 0167-4048,
[17] Hove, L.T. (2020). 'Strategies Used to Mitigate Social Engineering Attacks', Doctoral dissertation, Walden University.
[18] Huseynov, F. and Ozdenizci Kose, B. (2022) ' Using machine learning algorithms to predict individuals' tendency to be victim of social engineering attacks', Information Development, doi:https://doi.org/10.1177/02666669221116336.

[19] Lansley, M., Mouton, F., Kapetanakis, S. and Polatidis, N. (2020). SEADer++: social engineering attack detection in online environments using machine learning. Journal of Information and Telecommunication, 4(3), pp.346–362.
[20] Lansley, M., Polatidis, N., and Kapetanakis, S. (2019) 'SEADer: A Social Engineering Attack Detection Method Based on Natural Language Processing and Artificial Neural Networks', Lecture Notes in Computer Science, 11683, pp.686-696.
[21] Lansley, M., Polatidis, N., Kapetanakis, S., Amin, K., Samakovitis, G., and Petridis, M. (2019) 'Seen the villains: Detecting Social Engineering Attacks using Case-based Reasoning and Deep Learning', In Workshops Proceedings for the Twenty-seventh International Conference on Case-Based Reasoning, pp. 39-48,
[22] Li, J., Li, X. and Li, S. (2022) 'RoBERTa-based phishing email detection', Proceedings of the IEEE International Conference on Communications (ICC).
[23] Liu, K., Jain, A. and Liu, P. (2023) 'Continual learning for adaptive phishing detection', Proceedings of the ACM Symposium on Access Control Models and Technologies (SACMAT).
[24] Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G. and PRISMA Group, 2009. Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. PLoS Medicine, 6(7), p.e1000097.
[25] Mridha, K., Hasan, J., D, S. and Ghosh, A. (2021). Phishing URL Classification Analysis Using ANN Algorithm. [online] IEEE Xplore.
[26] Mughaid, A., AlZu'bi, S., Hnaif, A., Taamneh, S., Alnajjar, A. and Elsoud, E.A. (2022). An intelligent cyber security phishing detection system using deep learning techniques. Cluster Computing.
[27] Odeh, N.A., Eleyan, D. and Eleyan, A. (2021) 'A survey of social engineering attacks: detection and prevention tools', Journal of Theoretical and Applied Information Technology, 99 (18), pp. 4375 – 4386
[28] Raigada García, R.S. (2023) 'Application of Natural Language Processing (NLP) in Social Engineering: Emulating the Writing Style of a Hacked Subject Medium'
[29] Rathee, D. and Mann, S. (2022). Detection of E-Mail Phishing Attacks – using Machine Learning and Deep Learning. International Journal of Computer Applications, 183(47), pp.1–7.
[30] Ravi Kumar, G., Gunasekaran, S., Nivetha.R, Sangeetha Prabha.K, Shanthini.G and Vignesh.A.S (2019). URL phishing data analysis and detecting phishing attacks using machine learning in nlp. International Journal of Engineering Applied Sciences and Technology, 3(10), pp.26–31.
[31] Sahingoz, O.K. (2019) 'Phishing detection from URLs with deep learning', Expert Systems with Applications, 117, pp. 345–357.
[32] Sahoo, P., Mantri, S.K. and Kumar, D. (2018) 'Email phishing detection using RNN', Proceedings of the International Conference on Machine Learning and Data Engineering (iCMLDE).
[33] Sathvik, K., Gupta, P., Sitra, S.S., Subhashini, N. and Muthulakshmi, S. (2023) 'Social Engineering Attack Detection Using Machine Learning', in Chinara, S. et al. (eds) Advances in Distributed Computing and Machine Learning. Singapore: Springer, pp. 321–335.

[34] Sawa, Y., Bhakta, R., Harris, I.G. and Hadnagy, C. (2016). Detection of Social Engineering Attacks Through Natural Language Processing of Conversations. [online] IEEE Xplore.
[35] Syafitri, W., Shukur, Z., Mokhtar, U.A., Sulaiman, R. and Ibrahim, M.A. (2022). 'Social Engineering Attacks Prevention: A Systematic Literature Review', IEEE Access, [online] 10(1), pp.39325–39343. doi:https://doi.org/10.1109/ACCESS.2022.3162594.
[36] T. Mosa, D., Y. Shams, M., A. Abohany, A., M. El-kenawy, E.-S. and Thabet, M. (2023). 'Machine Learning Techniques for Detecting Phishing URL Attacks.' Computers, Materials & Continua, 75(1), pp.1271–1290. doi:https://doi.org/10.32604/cmc.2023.036422.
[37] Teixeira, M. n.d. 'Literature Review of Social Engineering Detection Models', University of Cape Town, DOI: 10.1145/1235
[38] Wang, Z., Ren, X., Li, S. and Zhang, J. (2022) 'Threat Detection for General Social Engineering Attack Using Machine Learning Techniques', arXiv preprint arXiv:2203.07933. Available at:
[39] Zhang, Z., Song, Y. and Sun, D. (2021) 'BERT-based phishing email detection with semantic augmentation', Neurocomputing, 432, pp. 42–52.
[40] Zvelo. (2024) 'The Role of AI in Social Engineering', [online] Available at: https://zvelo.com/the-role-of-ai-in-social-engineering/.