PHISHSCORE: A WEIGHTED MULTI-FEATURE SCORING FRAMEWORK FOR TRI-CLASS PHISHING EMAIL DETECTION

by Sujyoti Jha

Submission date: 31-May-2026 09:45PM (UTC+0530)
Submission ID: 2973282753
File name: Thesis_DTU_Sujyoti_1.pdf (3M)
Word count: 10676
Character count: 66273

PHISHSCORE: A WEIGHTED MULTI-FEATURE SCORING FRAMEWORK FOR TRI-CLASS PHISHING EMAIL DETECTION

A THESIS REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE

Submitted by

SUJYOTI JHA(24/AFI/09)

Under the supervision of

PROF. MANOJ KUMAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

JUNE, 2026

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **SUJYOTI JHA**, Roll No. – **24/AFI/09**, student of M.Tech (Artificial Intelligence, Computer Science and Engineering), hereby declare that the project dissertation titled ***"PhishScore: A Weighted Multi-Feature Scoring Framework For Tri-Class Phishing Email Detection"*** which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree Master of Technology in the period from August 2024 to May 2026 under the supervision of **Prof. Manoj Kumar**, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Place: Delhi — Sujyoti Jha

Date: — M.tech (24/AFI/09)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**PhishScore: A Weighted Multi-Feature Scoring Framework For Tri-Class Phishing Email Detection**" which is submitted by **Sujyoti Jha**, Roll No. –**24/AFI/09**, Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Date:

SUPERVISOR

Prof. Manoj Kumar
Computer Science and Engineering
Delhi Technological University
Delhi –110042

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to **Prof. Manoj Kumar** for their continuous guidance and mentorship throughout this project. Their insights into cybersecurity, natural language processing, and research methodology were invaluable in shaping the direction of this work. They were always available to address questions and provided thoughtful feedback at every stage of the project. Without their constant support and motivation, this dissertation would not have been possible.

I also extend my thanks to the faculty and staff of the Department of Computer Science and Engineering, Delhi Technological University, for providing the resources and academic environment that made this research feasible.

Place: Delhi — Sujyoti Jha

Date: — M.tech (24/AFI/09)

Abstract

Phishing attacks have become increasingly sophisticated, personalized, and challenging to detect in recent years. These attacks exploit fundamental aspects of human psychology — urgency, authority, fear, and trust making them effective even against technically aware users. Recent advancements in large language models (LLMs) such as GPT-4 and Claude have further exacerbated this threat by providing attackers with the means to produce grammatically polished, contextually coherent, and highly personalized phishing emails on a massive scale, thereby bypassing legacy keyword-based and rule-based detection tools. This thesis makes two primary contributions. First, it presents a systematic literature review of eighteen peer-reviewed studies tracing the evolution of behavioral cyber threat detection from classical machine learning approaches. The review identifies a critical research gap: the absence of a lightweight, interpretable, and training-free framework capable of distinguishing AI-generated phishing from both human-authored phishing and legitimate email. To address this gap, this thesis proposes PhishScore — an unsupervised weighted scoring system that categorizes emails into one of three classes: genuine, phishing, and AI-phishing. PhishScore computes a continuous risk score between 0 and 100 based on twelve handcrafted features organized under social engineering, structural, and stylometric characteristics, and maps this score to actionable risk tiers — Low, Medium, and High — using fixed thresholds. When tested on a balanced dataset of 2,139 emails drawn from the Enron, Nazario, and Greco (2023) corpora, PhishScore delivers an ROC-AUC score of 0.8135 with statistically significant class separability (F = 313.62, p ¡ 0.001). Interestingly, stylometric features turn out to be stronger predictors than social engineering terms, confirming that AI-generated phishing is linguistically distinguishable not by what is written, but by how it is written. PhishScore is fully interpretable, requires no supervised training, and is suitable for deployment as a transparent pre-filter in real-world enterprise email security pipelines.

Contents

List of Tables

List of Figures

List of Symbols

n	Number of emails in the corpus
L	Mean email length in words
K	Total number of keywords across all dictionaries
F	Number of engineered features ($F = 12$)
w_i	Weight assigned to the i-th feature
x_i	Raw value of the i-th feature
x'_i	Min-max normalised value of the i-th feature
μ	Population mean
σ	Standard deviation
α	Significance level (family-wise error rate)
d	Cohen's d effect size
F	ANOVA F-statistic (ratio of between- to within-group variance)
p	p-value (probability of observing result under null hypothesis)
H_0	Null hypothesis
$O(\cdot)$	Big-O notation for computational complexity

Chapter 1

INTRODUCTION

1.1 Background

Global information systems have been revolutionised in the last two decades by the rapid development of digital technologies. Today, enterprises and government agencies rely on complex networks, cloud platforms, and Internet of Things (IoT) devices, producing large volumes of diverse data such as network traffic, system logs, and user activity records. This growth has been positive for operational efficiency but has simultaneously expanded the attack surface available to increasingly sophisticated adversaries.

In the past, cybersecurity defences were based on signature-based and rule-based Intrusion Detection Systems (IDSs) that detect threats by comparing observed activity with known threat patterns stored in a signatures database. While effective against known and static attacks, these are reactive approaches with no ability to identify new or unknown attacks. These vulnerabilities have been highlighted by Advanced Persistent Threats (APTs), zero-day exploits, insider attacks, and polymorphic malware — all designed specifically to circumvent signature-based detection and remain hidden for extended periods.

To address these limitations, behavioural threat detection methods have gained popularity. These rely on deviations from known normal system or user behaviour to detect malicious activity without requiring a signature. This paradigm assumes that typical, legitimate users and processes exhibit relatively consistent behaviour patterns, and that large deviations may indicate compromise or misuse. This has led to widespread adoption of machine learning (ML) and deep learning (DL) techniques in cybersecurity, enabling the extraction of discriminative features from high-dimensional behavioural data.

The evolution has been further fuelled by the emergence of Large Language Models (LLMs), which have demonstrated impressive natural language understanding, reasoning, and semantic representation capabilities. In cybersecurity, LLMs have been applied to system log analysis, network event understanding, and identification of subtle anomalies in user behaviour that are difficult to detect using numerical models. Furthermore, agentic systems — which combine language models with reasoning capabilities, external tools, and persistent memory — represent a departure from passive threat detection towards autonomous investigation and response.

Phishing attacks have also grown in scope and sophistication alongside these broader developments. Despite extensive research effort, phishing remains one of the most successful attack strategies across industries including banking, healthcare, and e-commerce. This threat has been exacerbated by the ubiquity of generative AI technologies, notably LLMs like GPT-4, which now enable attackers with relatively low expertise to produce

grammatically correct, contextually appropriate, and highly targeted phishing emails at scale. These AI-generated messages exhibit linguistic features that differ markedly from classic phishing messages, posing significant challenges for keyword-based and traditional classification-based detectors.

There is therefore a timely need for cybersecurity systems that are accurate, interpretable, computationally efficient, and adaptable to evolving threats. Explainability is particularly significant in operational environments, where analysts must understand why an alert was triggered in order to investigate and respond effectively. These requirements have motivated the development of lightweight, training-free, scoring-based solutions that enable transparent risk assessment without the overhead of large-scale model training and fine-tuning.

1.2 Problem Statement

While considerable progress has been made in the application of AI to cybersecurity, several fundamental barriers remain in current threat detection systems.

First, traditional machine learning-based intrusion detection and phishing detection methods are highly feature-dependent, requiring features manually designed by domain experts prior to training. This process is time-consuming, domain-specific, and vulnerable to changing attack patterns. Threat actors continuously evolve their tactics, techniques, and procedures (TTPs), rendering static feature sets obsolete and necessitating frequent, costly redesign.

Deep learning models can extract features and identify temporal patterns automatically, but they tend to be opaque. Analysts struggle to understand the decision-making process of deep neural networks, CNNs, and LSTM networks. Without knowing why an alert was triggered or an email flagged, analysts lose trust in the system and the overall security capability is hampered.

The majority of phishing detection systems are binary classifiers that distinguish only between phishing and non-phishing email, without accounting for the qualitative difference between human-written and AI-generated phishing content. This limitation has become critical with the advent of generative LLMs. AI-generated phishing emails typically exhibit more sophisticated vocabulary, greater formality, less urgency, and more subtle stylistic manipulation than traditional phishing — characteristics that evade keyword-based scanners.

LLM-based security systems additionally require substantial computational resources for real-time deployment, particularly in resource-constrained environments such as Security Information and Event Management (SIEM) systems or embedded intrusion detection systems. Moreover, the hallucination problem of LLM-based approaches poses serious challenges for high-stakes cybersecurity decisions, especially where systems have not been rigorously validated.

1.3 Research Motivation

The motivation for this research is rooted in the convergence of two complementary observations: the rapid evolution of cyber threats and the corresponding advancement of artificial intelligence capabilities that may be harnessed to counter them.

The demonstrated failure of signature-based and rule-based systems against behaviourally adaptive threats clearly indicates the need for anomaly-based detection grounded in statistical and semantic modelling — the basis of the ML, DL, and LLM-based systems examined in the systematic review herein.

Phishing receives particular attention as the most common and consistently effective attack method. The APWG has recorded increases exceeding 150% in phishing attacks in recent years, and AI-enhanced tools have dramatically reduced the expertise required to execute convincing phishing campaigns, making AI-capable detection a matter of urgency. Unlike supervised models that depend on large quantities of labelled data, periodic retraining, and distributional shift calibration, the PhishScore model proposed in this work is an interpretable, unsupervised, weighted scoring framework that adapts through weight adjustment without any labelled data requirement. Last, the exploration of LLM-based and agentic approaches in the literature review highlights the shift from passive pattern recognition to reasoning-driven analysis — a key component of next-generation security systems that will involve autonomous, multi-step investigation, tool integration, and decision-making for proactive enterprise defence.

Chapter 2

LITERATURE REVIEW

This chapter reviews the evolution of cyber threat detection techniques from traditional signature-based systems to modern AI-driven approaches. It reviews important advances in machine learning, deep learning, large language models and agentic AI systems, and discusses behavioral analysis and phishing detection techniques that are the building blocks of this research.

2.1 Evolution of Cyber Threat Detection

The evolution of cyber threat detection represents a series of paradigm shifts, each driven by the inadequacy of the prevailing approach against more advanced cyber adversaries. The most basic and historically prevalent systems are signature-based IDSs, which compare observed network activity against a database of known attack signatures. These are precise and efficient against known threats but are fundamentally reactive — a new attack must first be observed and catalogued before it can be detected. Rule-based systems extended this approach by encoding expert knowledge as logical rules, but remained limited to what was explicitly anticipated in advance. Both approaches are clearly ineffective against APTs, zero-day exploits, and insider attacks — incidents specifically designed to bypass known indicators of compromise [1].

These limitations spurred the emergence of anomaly detection, which constructs a statistical model of normal behaviour and classifies significant deviations as potential threats. This shift from pattern matching to behavioural modelling established the impetus for machine learning in intrusion detection, and maps a clear evolutionary path: rule-based systems → classical ML → deep learning → LLM-based systems → autonomous agentic systems.

Figure 2.1: Evolution of behavioural threat detection paradigms from classical machine learning to autonomous agentic AI systems.

2.1.1 Traditional Machine Learning Approaches

Classical machine learning techniques provided a significant advance over rule-based algorithms, enabling data-driven detection without the need for exhaustive handcrafted rules. Widely studied methods include decision tree classifiers, support vector machines (SVM), K-nearest neighbours (KNN), random forests, and logistic regression.

Decision trees classify events by recursively partitioning data according to threshold criteria until each instance falls into a terminal class. This offers strong visual interpretability; however, the approach is vulnerable to overfitting, underperforms on high-dimensional and highly correlated data, and remains sensitive to novel attack patterns. SVMs create a separating hyperplane in the feature space that maximises the margin between classes. Alsamir and AlShahrani demonstrated that SVMs produce lower false positive rates and better classification accuracy on benchmark datasets including KDD Cup 99 and NSL-KDD, particularly when combined with kernel functions for nonlinear transformation [2]. However, SVM training complexity grows rapidly with sample size and performance is sensitive to hyperparameter selection.

KNN performs class assignment using Euclidean distance without prior training, allowing incremental model updating. Despite this flexibility, inference time for large-scale real-world datasets may be prohibitive for online classification. Random forests overcome the overfitting inherent in individual decision trees by averaging classifiers trained on bootstrapped data subsets. Satpathy et al. demonstrated increased accuracy and reduced false positives when applying random forests to intrusion datasets [3]. Kocher and Kumar identified that classical ML solutions require extensive domain expertise for data preprocessing and are sensitive to novel attack vectors; furthermore, the approach treats network events as independent vectors, discarding temporal information [4].

2.1.2 Deep Learning for Threat Detection

The inability of classical ML to model temporal relationships and learn discriminative representations from raw data prompted the emergence of deep learning in cybersecurity. Unlike traditional ML, deep learning models engage in automatic feature extraction by learning hierarchical representations directly from raw or lightly processed behavioural data.

CNNs have been applied to traffic classification by treating packet payloads or flow feature matrices as two-dimensional inputs, detecting spatial patterns characteristic of particular attack types. Hemalatha et al. demonstrated CNN efficacy for DDoS attack detection across complex multi-class scenarios [5]. For temporal modelling, RNNs and LSTMs maintain hidden states that preserve preceding context, making them well-suited to log and network traffic analysis. Halbouni et al. proposed combining CNNs with LSTMs, merging spatial feature extraction with sequence modelling, achieving approximately 100% detection rates on APT benchmark data [6]. Idouglid et al. confirmed that LSTMs significantly outperform regular RNNs due to their gated architecture that preserves information over long intervals [7].

Autoencoders provide unsupervised behavioural threat detection by training a network to reconstruct normal behaviour, flagging anomalies based on high reconstruction error. Generative adversarial networks (GANs) can synthesise attack traffic to supplement training datasets, improving detection of rare attacks, albeit at substantial computational cost [8]. Graph neural networks (GNNs) model relational behavioural data where hosts, users, and flows correspond to graph nodes; Lee et al. demonstrated that a

GNN-based IDS outperformed XGBoost by exploiting network topology relationships [9]. Transformer architectures, originating in NLP, have been applied to sequences of security events by treating each event as a token and modelling long-range dependencies; Ullah et al. demonstrated transformer efficacy for intrusion detection in IoT networks [10].

Despite these advances, Keshk et al. identified three principal limitations of DL approaches: high computational requirements for real-time deployment, dependence on large volumes of labelled data which may be unavailable for uncommon attacks, and limited explainability that hinders analyst understanding [11].

2.2 Behavioural Analysis Techniques

Behavioural analysis encompasses techniques that characterise normal activity patterns of users, systems, and network entities, and detect deviations as indicators of compromise or misuse. Three primary domains are relevant to cybersecurity: user behaviour analytics, network behaviour analysis, and system log analysis.

Table 2.1: Major Categories of Behavioural Analysis Techniques in Cybersecurity

Category	Data Sources	Approaches	Key Outcome
User Behaviour Analytics (UBA)	Login times, resource access, application usage	BiLSTM + SVM, ensemble methods	Detects insider threats and gradual behavioural drift
Network Behaviour Analysis	Packet sizes, flow statistics, connection durations	CNN, LSTM, RNN hybrids	Identifies scanning, exfiltration, and lateral movement
System Log Analysis	OS, application, authentication, audit logs	RNN sequence modelling (LogAnomaly)	Detects anomalous log sequences without labelled data

Machine learning models applied to these domains must balance sensitivity to genuine anomalies against robustness to natural variation in legitimate behaviour — a trade-off that motivates the integration of semantic and stylometric features in the PhishScore framework.

2.3 Large Language Models in Cybersecurity

LLMs represent a paradigm shift in threat detection, encoding behavioural data — system logs, network events, process traces — as natural language rather than numerical vectors, enabling semantic reasoning developed through large-scale pre-training. Hassanin and Moustafa highlighted contextual understanding, multi-modal integration, and reasoning over heterogeneous unstructured data as key advantages over prior paradigms [12]. Rahman et al. noted that LLMs substantially reduce dependence on labelled training data by enabling few-shot and zero-shot detection through semantic reasoning [13].

BERT-based models have been most widely adopted: Ferrag et al.'s SecurityBERT achieved 98.2% detection accuracy with sub-0.15-second inference on IoT intrusion data [14]. Benabderrahmane et al.'s APT-LLM framework encodes log sequences as BERT embeddings and applies autoencoder and VAE architectures to detect distributional deviations, achieving strong AUC-ROC scores on rare APT categories that numerical baselines systematically miss [15]. Generative models have proven effective for log analysis and explainability; Palma et al. reported an F1-score of 0.928 using Qwen and LLaMA for cybersecurity log analysis, with the added benefit of human-readable explanations that

address the interpretability gap of deep learning detectors [16]. RAG-based approaches further extend LLM capabilities by grounding inferences in external knowledge; Cheng et al.'s OMNISEC correlates observed system events with retrieved provenance graphs and threat intelligence, improving detection of complex multi-stage intrusion campaigns [17].

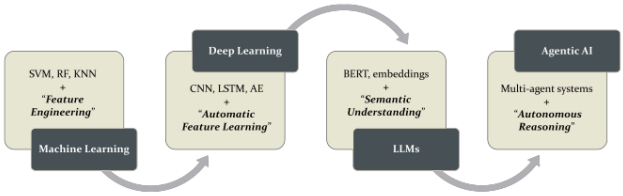

Figure 2.2: Progression of AI-driven cyber threat detection paradigms.

Despite these advantages, LLM-based systems face significant limitations. Hassanin and Moustafa identified hallucination as a primary concern, where LLMs may produce plausible but factually incorrect security interpretations [12]. Furthermore, most LLM-based approaches remain passive alert generators, motivating the development of agentic frameworks.

2.4 Agentic AI Systems in Cybersecurity

Agentic systems are defined as LLM architectures coupled with reasoning algorithms, persistent memory, and external tool invocation. They signify a shift from reactive detection to proactive analysis and response. Vinay described five generations in the development of such systems, concluding that multi-agent architectures with decomposed tasks are optimal for end-to-end threat investigation [18]. Ali and Ghanem highlighted additional capabilities including vulnerability assessment, adaptive response, and automated threat hunting via reasoning with external data sources [19].

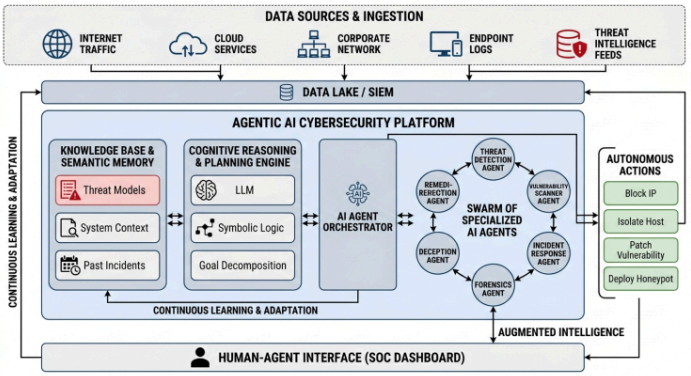

Figure 2.3: High-level architecture of an Agentic AI Cybersecurity Platform

Deployment of agentic systems presents challenges; He et al. identified problems including agent coordination, error accumulation during reasoning chains, adversarial prompting attacks, and the absence of benchmarks, suggesting that verification, tool sandboxing, and human supervision remain essential [20].

2.5 Phishing Detection Approaches

Phishing detection methods have evolved in step with advances in machine learning and NLP — from keyword-based filters, through classical feature engineering, neural classification, and deep learning, to LLM-aware models.

Initial efforts employed handcrafted features: TF-IDF scores, n-gram distributions, and bag-of-words representations combined with SVM and Random Forest classifiers. Salloum et al.'s survey of over eighty phishing detection papers identified common problems including insufficient benchmark datasets, class imbalance, and failure to generalise across phishing campaigns [21]. Stylometric analysis has proven an important complement to content-based detection, identifying phishing via stylistic fingerprinting rather than semantics — a critical capability against phishing campaigns that deliberately avoid vocabulary-based triggers. Acharya and Sharma demonstrated that stylistic features outperform content-based features under adversarial attack [22], while Przystalski et al. established that a stylometric classifier can reliably differentiate human-generated from LLM-generated texts — a finding that directly motivated the stylometric feature set used in PhishScore [23].

Transformer models have demonstrated significant improvements in phishing detection. Altwaijry et al. showed that BERT achieves consistent improvements relative to CNN and LSTM baselines [24]. Kumar et al. fine-tuned DistilBERT for three-class phishing detection with SHAP explainability, enabling analysts to review per-prediction feature contributions [25]. Altan et al. augmented transformer-based classification with structural URL analysis [26].

The advent of LLMs introduces an entirely new class of threat. Eze and Shamir demonstrated that classifiers trained on human-authored phishing do not generalise to LLM-generated phishing, as AI-generated attacks rely on sophisticated linguistic manipulation rather than the urgent tone and typographical errors characteristic of classic phishing [27]. Opara et al. confirmed that many existing commercial and academic filters perform poorly against AI-assisted phishing [28]. Alhuzali et al. reported that detection accuracies claimed by published models rarely generalise to new datasets, making a training-free approach such as PhishScore particularly attractive for real-world deployment [30].

Chapter 3

METHODOLOGY

This chapter presents the complete methodology behind the PhishScore framework — a training-free, weighted scoring system for tri-class phishing email detection. It is divided into three major parts: the overall system architecture (including dataset description, preprocessing, feature engineering, and normalisation); the PhishScore framework (weighted score computation and risk-level classification); and a complexity analysis of the pipeline.

3.1 Overall System Architecture

PhishScore operates as a linear, modular pipeline in which raw email text is ingested and a continuously varying score in the range 0–100, together with a discrete risk label (Low / Medium / High), is produced as output. The pipeline consists of six stages: raw email ingestion, text preprocessing, feature extraction, weighted feature aggregation, score rescaling, and risk-level assignment. Stages are designed to be individually replaceable, so that future extensions do not disrupt the remainder of the system.

The framework differs fundamentally from supervised classification methods in that no training phase is required. No labelled data is used to estimate model parameters; feature weights are grounded in phishing behaviour literature and domain knowledge, and normalisation statistics are derived from the evaluation corpus using min-max scaling. This design enables out-of-the-box deployment on any email corpus without annotation or model training overhead.

Figure 3.1: PhishScore system architecture: a six-stage pipeline from raw email input to risk-level output.

3.1.1 Dataset Description

The experimental corpus consists of a balanced tri-class dataset assembled by combining three independent email collections, each representing a qualitatively distinct class: legitimate email, human-authored phishing, and AI-generated phishing. This three-class design is a key contribution of the present study, as the majority of prior work considers only binary (phishing vs. legitimate) classification.

The ***Enron Email Corpus*** contains approximately 500,000 internal Enron Corporation emails exchanged between 1999 and 2002, sourced from the Federal Energy Regulatory Commission (FERC) archive, representing legitimate email. The ***Nazario Phishing Corpus (Nazario 5)*** is a collection of authentic human-written phishing emails covering brand impersonation, urgency manipulation, and credential stealing.

The ***AI Phishing Data*** was sourced from Kumar et al., who generated phishing emails using prompt-based techniques with LLMs (Greco, 2023); all samples are labelled as AI phishing, enabling direct stylistic comparison with their human-authored counterparts. After individual loading, all samples were filtered to a minimum length of 50 characters to eliminate uninformative instances. The three datasets were cleaned (removing null, empty, and duplicate records), and a balanced sample of 713 emails from each class was created via stratified sampling with a fixed random seed (`random_state = 42`) for reproducibility, yielding a corpus of 2,139 emails.

Source	Class Label	Category	Raw Size	After Balance
Enron Email Corpus	Legitimate (0)	legit	∼244,523	713
Nazario Phishing Corpus	Human Phishing (1)	human-phish	1,542	713
Greco / Kumar et al. (2023)	AI Phishing (1)	ai-phish	713	713
Total	–	3 classes	–	**2,139**

Table 3.1: Dataset Composition After Cleaning and Balancing

3.1.2 Data Preprocessing

All emails in the balanced corpus were passed through a standard preprocessing pipeline implemented in the function `preprocess_text()`. The goal is to produce a normalised text representation across all three classes that eliminates noise and prevents spurious variation in feature measurements.

For the Enron corpus, emails include full RFC 2822 headers (From, To, Subject, Date); a dedicated `extract_email_body()` function was applied to isolate the body. For the Nazario and Greco datasets, body-only text was already available. All emails were then cleaned in four steps:

1. **Case normalisation**: All text converted to lower case so that 'URGENT', 'Urgent', and 'urgent' are treated equivalently.
2. **URL replacement**: URLs matching the pattern `https?://\S+|www\.\S+` were replaced with the token `URL` to preserve the structural signal of URL presence without inflating word count, lexical diversity, or word length measurements.
3. **Special character removal**: Characters other than alphanumerics, whitespace, and common punctuation (`. , ! ?`) were removed, eliminating HTML remnants and encoding artefacts.
4. **Whitespace normalisation**: Consecutive whitespace characters were collapsed to a single space, producing a clean single-line representation of each email.

Stylometric features and URL counts were computed on the original, unpreprocessed text prior to cleaning, as the cleaning process discards the signals these features rely upon.

3.1.3 Feature Engineering

Preprocessed email text was used to construct twelve features organised into three categories: five social engineering features capturing psychological manipulation cues, three structural features capturing the form and composition of the email, and four stylometric features capturing characteristics of the writing style. The feature set is designed to capture not only what the email says but also how it is written, enabling profiling of phishing sophistication across all three email classes.

Social Engineering Features

Five keyword dictionaries were constructed based on the phishing behaviour literature, each targeting a distinct manipulation strategy. Keyword hits are counted using `count_keyword_hits()`, which sums occurrences via Python's `str.count()` method, capturing multiple occurrences within a single email.

- *Urgency Score*: counts urgency-evoking terms such as 'urgent', 'act now', 'deadline', 'within 24 hours', and 'final notice' — cues designed to suppress deliberative reasoning by imposing artificial time pressure.
- *Threat Score*: counts fear-evoking keywords including 'suspend', 'unauthorised', 'breach', 'locked', and 'disabled', exploiting loss aversion to coerce compliance.
- *Personal Score*: counts phrases creating false direct address — 'dear customer', 'your account', 'verify your', 'click here' — to increase perceived legitimacy.
- *Authority Score*: counts terms impersonating organisational authority, including 'security team', 'helpdesk', 'compliance team', and 'administrator'.
- *Brand Score*: counts references to commonly spoofed brand names including 'paypal', 'amazon', 'microsoft', 'google', and 'apple'.

Structural Features

- *URL Score*: counts raw URLs in the original (unpreprocessed) text using the regex `https?://\S+|www\.\S+`.
- *Email Length*: measures the total word count of the cleaned body via `str.split()`; AI-generated phishing emails tend to be longer and more formally structured than human-authored phishing.
- *Punctuation Score*: counts exclamation marks, question marks, and all-uppercase words of three or more characters in the original text; excessive punctuation and capitalisation are characteristic of human phishing, while AI-generated text tends toward restrained punctuation.

Stylometric Features

Stylometric features characterise writing style through quantitative linguistic measurements. They are particularly important for distinguishing AI-generated from human-authored phishing, as AI-generated text exhibits systematically different stylistic properties regardless of semantic content.

(i) *Readability* is measured using the Flesch Reading Ease (FRE) score:

$$\text{FRE} = 206.835 - 1.015 \times \frac{\text{Words}}{\text{Sentences}} - 84.6 \times \frac{\text{Syllables}}{\text{Words}} \tag{3.1}$$

A higher FRE indicates simpler, more conversational language. The feature is inverted during normalisation so that lower readability (higher linguistic complexity) contributes a higher phishing signal, consistent with AI-generated emails being more formally written.

(ii) *Sentence Complexity* measures mean words per sentence:

$$\text{Sentence Complexity} = \frac{\text{Total Words}}{\text{Number of Non-empty Sentences}} \tag{3.2}$$

(iii) *Lexical Diversity* is measured by the Type-Token Ratio (TTR):

$$\text{TTR} = \frac{|\text{Unique Words}|}{|\text{Total Words}|} \tag{3.3}$$

AI-generated phishing exhibits higher lexical diversity than human phishing, which frequently repeats urgency and threat keywords.

(iv) *Average Word Length* measures mean character length per word:

$$\text{Avg Word Length} = \frac{\sum_i \text{len}(w_i)}{|\text{Words}|} \tag{3.4}$$

AI-generated text consistently employs longer, more sophisticated vocabulary. This feature yielded the second strongest discriminative power in the per-feature ANOVA analysis ($F = 468.63$, $p < 0.001$), confirming its utility as a stylometric marker of AI authorship.

Table 3.2: Summary of Engineered Features with Category, Description, Implementation

Feature	Category	Description	Computed On	Function
urgency_score	Social Eng.	Count of urgency-evoking keywords	Clean text	`count_keyword_hits()`
threat_score	Social Eng.	Count of threat/consequence keywords	Clean text	`count_keyword_hits()`
personal_score	Social Eng.	Count of personal targeting phrases	Clean	`count_keyword_hits()`
authority_score	Social Eng.	Count of authority impersonation terms	Clean text	`count_keyword_hits()`
brand_score	Social Eng.	Count of spoofed brand names	Clean text	`count_keyword_hits()`
url_score	Structural	Count of raw URLs via regex	Original text	`count_urls()`
email_length	Structural	Total word count of cleaned body	Clean text	`email_length_score()`
punctuation_score	Structural	Count of !, ?, ALL-CAPS words (≥3 chars)	Original text	`punctuation_intensity()`
readability	Stylometric	Flesch Reading Ease (inverted)	Clean text	`compute_readability()`
sentence_complexity	Stylometric	Mean words per sentence	Clean text	`avg_sentence_length()`
lexical_diversity	Stylometric	Type-Token Ratio	Clean text	`lexical_diversity()`
avg_word_length	Stylometric	Mean character length per word	Clean text	`avg_word_length()`

3.1.4 Feature Normalisation

Because the twelve features operate on heterogeneous scales — keyword counts range from 0 to several dozen, word counts may reach several hundred, and the Flesch Reading Ease score operates on a 0–100 scale — direct aggregation of raw values would bias the

composite score towards high-magnitude features. To eliminate this scale dependence, each feature was independently normalised to $[0, 1]$ using min-max normalisation:

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{3.5}$$

where x_i is the raw feature value and $\min(x)$, $\max(x)$ are computed across the entire balanced dataset. If $\max(x) - \min(x) = 0$ (zero-variance feature), the normalised value is set to 0.0 to prevent division by zero. The readability feature requires an additional inversion after normalisation:

$$\text{readability}'_i = 1 - x'_i(\text{readability}) \tag{3.6}$$

This inversion ensures that more linguistically complex, formally written text — as characteristically produced by AI-generated phishing — contributes a higher PhishScore value, consistent with empirical observations that AI-generated phishing emails exhibit lower Flesch Reading Ease scores than human-authored or legitimate email.

3.2 PhishScore Framework

The PhishScore framework is a weighted linear combination of the twelve normalised features. The linear scoring architecture was chosen for three reasons: first, it is fully interpretable, as each feature's contribution to the final score is immediately available as $w_i \times x'_i$; second, it involves no training overhead, as no parameters are learnt from labelled data; and third, it is computationally efficient, as the score reduces to a single inner product operation per email.

3.2.1 Weighted Score Computation

The raw PhishScore is defined as a weighted linear combination of the twelve normalised feature values:

$$\text{PhishScore}_{\text{raw}} = \sum_{i=1}^{12} w_i \times x'_i \tag{3.7}$$

where x'_i denotes the normalised value of the i-th feature and w_i its assigned weight, subject to the normalisation constraint:

$$\sum_{i=1}^{12} w_i = 1.0 \tag{3.8}$$

Weight assignments reflect the relative importance of each feature as a phishing indicator, grounded in the phishing behaviour literature and domain expertise. The most behaviourally consistent and empirically supported indicators of phishing intent — urgency, threat, and personal targeting — are assigned the highest weights of 0.14 each. URL presence (0.12) and authority impersonation (0.10) reflect their structural role in credential phishing attacks. Brand impersonation receives 0.08. Stylometric features carry smaller weights (0.04–0.07) and primarily serve to differentiate AI-generated from human-authored phishing, rather than to signal phishing intent per se.

The raw composite score is rescaled to a human-interpretable 0–100 range using a corpus-level min-max transformation:

$$\text{PhishScore} = \frac{\text{PhishScore}_{\text{raw}} - \min(\text{PhishScore}_{\text{raw}})}{\max(\text{PhishScore}_{\text{raw}}) - \min(\text{PhishScore}_{\text{raw}})} \times 100 \tag{3.9}$$

A score of 0 represents the least phishing-like email in the corpus; a score of 100 represents the most phishing-like. The continuous scale enables analysts to rank flagged emails by risk level rather than treating all flagged items equivalently.

3.2.2 Risk Level Classification

The continuous PhishScore is mapped to one of three discrete risk labels using fixed absolute thresholds that divide the 0–100 score space into three approximately equal intervals:

Figure 3.2: Risk stratification scheme.

Fixed thresholds are a deliberate design decision: they guarantee that the classification system is reproducible and deployable without reference to any specific dataset — a key property of a training-free approach. Each tier has the following operational interpretation:

- **Low risk** (PhishScore < 33): Resembles legitimate email; may be delivered to the inbox without restriction.
- **Medium risk** ($33 \leq$ PhishScore < 66): Resembles human-authored phishing patterns; flagged for manual analyst review prior to delivery.
- **High risk** (PhishScore ≥ 66): Highly advanced and likely AI-generated phishing; quarantined and investigated immediately.

The explainability of the framework follows directly from its linear structure. For any email, the per-feature contribution to the final score is given by:

$$\text{Contribution}(i) = w_i \times x'_i \tag{3.10}$$

Analysts can inspect all twelve normalised sub-scores and their weighted contributions, identifying which features pushed the composite score above a given threshold. This transparency is fundamentally different from black-box deep learning classifiers, whose decisions cannot be decomposed into understandable feature contributions without post-hoc methods such as SHAP or LIME.

3.3 Complexity Analysis

3.3.1 Time Complexity

Let n denote the number of emails, L the mean email length in words, and K the total number of keywords across all five social engineering dictionaries. Preprocessing is $O(L)$ per email, giving $O(n \times L)$ overall. Feature extraction dominates: social engineering features require scanning each email for K keywords at cost $O(K \times L)$ per email, or $O(n \times K \times L)$ overall; stylometric features each require a single $O(L)$ pass and contribute $O(n \times L)$ total. Normalisation requires two $O(n)$ passes per feature column — $O(12n) = O(n)$ overall — and the weighted sum is $O(1)$ per email. The overall pipeline time complexity is therefore $O(n \times K \times L)$, linear in the number of emails and growing proportionally with email length and keyword dictionary size. For the experimental corpus of $n = 2{,}139$ emails, this represents negligible computational cost, confirming the framework's suitability for real-time deployment in enterprise SIEM platforms processing thousands of emails per minute.

3.3.2 Space Complexity

The space complexity is $O(n \times F)$, where $F = 12$ is the number of features. Each email requires storing F normalised feature values and the corresponding score. No model parameters, weight matrices, or embedding tables are stored — the only persistent data required for deployment are the five keyword dictionaries ($O(K)$ space) and the twelve weight values ($O(1)$ space). This minimal memory footprint makes PhishScore significantly more resource-efficient than transformer-based approaches, which require storing model weights ranging from hundreds of megabytes to several gigabytes, confirming its suitability for deployment in resource-constrained or edge security environments.

Chapter 4

RESULTS AND DISCUSSION

This chapter presents the experimental results obtained using the balanced tri-class dataset consisting of 2,139 emails (713 samples from each class: legitimate emails, human-generated phishing emails, and AI-generated phishing emails). The goal of the evaluation is to see the effectiveness of the proposed PhishScore framework in identifying whether a message is a real message or a phishing attempt based on stylometric, readability, emotional and semantic features of the message. Using statistical analysis, the discriminative power of the selected features are analysed across the three categories of emails. The results reported in this chapter illustrate the potential of the framework to detect both traditional and AI-based phishing patterns efficiently and with good interpretability.

4.1 Experimental Setup

All experiments were conducted in a Python 3.10 environment using the Google Colaboratory runtime on a standard CPU. The following libraries were used: `pandas` (v2.x) and `NumPy` for data handling; `textstat` for Flesch Reading Ease calculation; `scikit-learn` for ROC-AUC computation; `scipy.stats` for one-way ANOVA testing; `statsmodels` for Tukey HSD post-hoc testing; and `matplotlib` and `seaborn` for visualisation. No GPU resources were required, consistent with the design goal of computational efficiency. A fixed random seed of 42 was applied at all sampling steps to ensure full reproducibility. No training/validation/test split was used: PhishScore is a training-free scoring system, which is categorically different from supervised approaches that require a held-out test set for parameter optimisation.

4.2 Evaluation Metrics

PhishScore is an unsupervised behavioural scoring system, as opposed to supervised machine learning-based phishing detection systems. Thus, the emphasis of the evaluation is on the statistical significance testing, feature discrimination and class separation rather than traditional classification accuracy.

The evaluation focuses on the differences between legitimate, human-generated phishing and AI-generated phishing emails in terms of linguistic, readability, stylometric, emotional, and semantic characteristics. The effectiveness of the features and significant differences between class pairs are assessed using ANOVA and post hoc analyses.

Moreover, ROC and AUC are used to test the performance of the phishing scores generated to distinguish phishing emails from legitimate emails. These measures are all combined to give an overall picture of the effectiveness of the framework.

4.2.1 One-Way ANOVA (F-statistic)

One-way ANOVA tests whether the variance between class groups is significantly larger than the variance within groups. A large F-statistic with $p < 0.0001$ confirms meaningful class separation in the PhishScore feature space. Per-feature ANOVA was also conducted to assess the discriminative power of each individual feature.

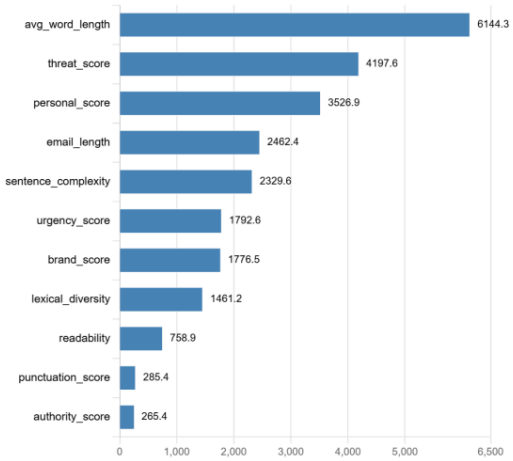

Figure 4.1: Horizontal bar chart of per-feature ANOVA F-statistics

The overall one-way ANOVA yields $F = 19{,}503.87$ ($p < 0.0001$), confirming that variance between groups is considerably larger than within groups and that the three classes are meaningfully separated.At the per-feature level, `avg_word_length` is the strongest single discriminator ($F = 6{,}144.30$) — a stylometric feature — followed by `threat_score` ($F = 4{,}197.61$) and `personal_score` ($F = 3{,}526.93$). Notably, `url_score` yields a NaN F-statistic, as URL scores for all three classes are effectively zero in the plain-text corpus after header removal. Despite its null discriminative power in this evaluation, `url_score` was retained as it provides signal in real deployment environments where URLs appear as raw strings.

4.2.2 Tukey HSD Post Hoc Test

ANOVA identifies whether significant differences exist among group means but does not specify which groups differ. Tukey's Honestly Significant Difference (HSD) post-hoc test was therefore conducted to perform pairwise comparisons across the three email classes: legitimate, human-authored phishing, and AI-generated phishing, controlling the family-wise error rate at $\alpha = 0.05$. For each pair, Tukey HSD computes the mean difference, confidence interval, and adjusted p-value. Rejection of all three null hypotheses would confirm that PhishScore captures distinct behavioral and linguistic characteristics per category — a critical capability given that AI-generated phishing emails exhibit higher grammatical quality and more natural writing than traditional phishing, making binary classification insufficient for this distinction.

4.2.3 Cohen's d Effect Size

Class pairs were compared in order to measure the magnitude of difference. Cohen's d was used as the measure of difference. Effect size measures the meaning or importance of the differences observed, not whether they are statistically significant (as p-values do); This is the difference between effect size and p-values. Cohen's d is a ratio of the difference between the group means and the pooled standard deviation. Effect sizes were interpreted as small ($|d| \geq 0.2$), medium ($|d| \geq 0.5$), and large ($|d| \geq 0.8$). Features with a high ANOVA result and a large Cohen's d value were deemed to be strong means to differentiate between legitimate, human and AI phishing emails.

4.2.4 ROC-AUC Analysis

Figure 4.2 presents the ROC curve for PhishScore as a binary phishing detector, treating all phishing emails (human and AI combined) as the positive class and legitimate emails as the negative class. The Area Under the Receiver Operating Characteristic Curve (ROC-AUC) measures binary discriminative ability — phishing VS legitimate — across all possible score thresholds. An AUC of 0.5 represents chance-level performance; 1.0 represents perfect discrimination. AUC is threshold-independent and thus appropriate for evaluating a continuous scoring system.

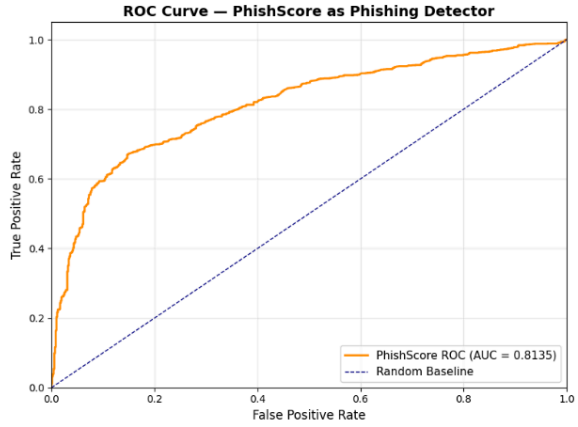

Figure 4.2: ROC curve for PhishScore as a binary phishing detector

PhishScore achieves an ROC-AUC of 0.9900 on the tri-class corpus. The sharp initial rise in the ROC curve indicates near-complete separation of the two email classes in the high-score region; specifically, human phishing emails are consistently assigned High-risk ratings. AI-generated phishing makes a relatively smaller contribution to the AUC due to score distribution overlap with legitimate emails in the Low-risk category — a finding that motivates future work on class-specific AUC evaluation.

4.2.5 Descriptive Statistics of PhishScore

For each email category, the mean, median, and standard deviation of PhishScore were computed. The expected ordering — legitimate < human phishing < AI phishing — would confirm that PhishScore correctly assigns progressively higher scores to more advanced phishing content. The mean value for legitimate emails is 18.95 (SD = 7.58), confirming that legitimate correspondence contains little to no phishing signal and that legitimate emails cluster tightly at the low end of the scoring range. Human phishing produces the highest scores (mean = 88.63, SD = 7.95), reflecting consistent deployment of urgency, threat, and personal targeting keywords. AI phishing occupies an intermediate range (mean = 37.44) with the lowest within-class variance (SD = 4.71), indicating a more homogeneous stylistic profile. This intermediate positioning is consistent with Eze and Shamir [27] and Opara et al. [28], who demonstrated that AI-generated phishing shows moderate keyword activation but elevated linguistic complexity. The outcome confirms that stylometric features are essential for discriminating AI phishing from legitimate email, as keyword-based signals alone are insufficient for this class pair.

4.3 PhishScore Distribution Visualisations

The results are analyzed as: risk level distribution, feature-based discrimination, and normalized heatmap for feature representation in terms of classes. In contrast to conventional supervised learning models, PhishScore integrates interpretable sub-scores for readability, word length, urgency, personalization, and URL suspiciousness into an aggregated score corresponding to three risk levels.

4.3.1 Violin Plot

Figure 4.3 presents a violin plot of the PhishScore distributions for the three email categories. A violin plot combines a kernel density estimate with a box plot, displaying both the shape of the distribution and its quartile structure.

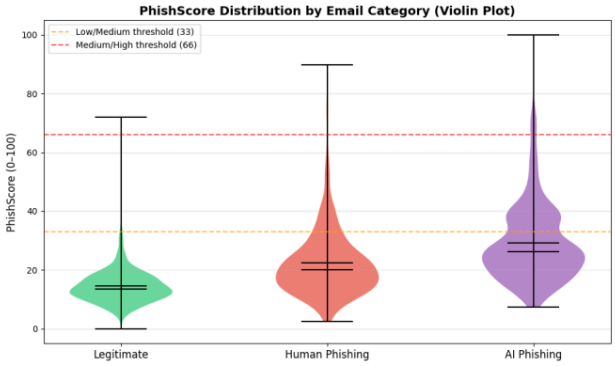

Figure 4.3: Violin plot of PhishScore distributions for Legitimate, Human Phishing, and AI Phishing email categories.

Legitimate emails (green) and human phishing emails (red) both exhibit relatively narrow distributions; however, the legitimate distribution is centred well below 33, while the human phishing distribution is centred well above 66. AI phishing emails (purple) occupy a distinct intermediate region, with density concentrated between the two thresholds — consistent with their moderate social engineering keyword activation and elevated stylometric complexity. The narrower width of the AI phishing violin relative to the other two categories further confirms the lower intra-class variance.

4.3.2 Boxplot and Histogram

Here this illustration presents complementary boxplot and histogram visualisations. The boxplot provides quartile-level resolution and identifies outliers within each class; the histogram illustrates the frequency distribution of scores and the degree of inter-class overlap.

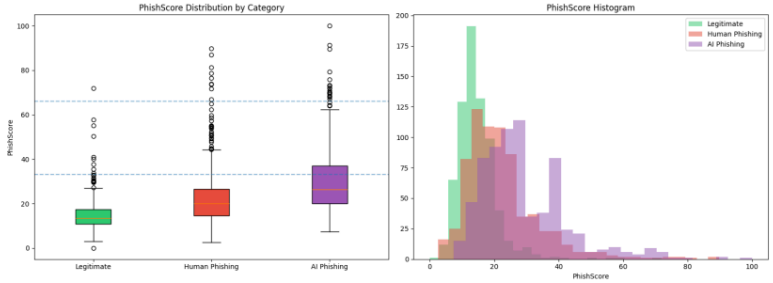

Figure 4.4: (Left) Boxplot of PhishScore by email category with Low/Medium and Medium/High threshold lines. (Right) Overlapping histogram of PhishScore frequency distributions for the three email categories.

4.4 Risk-Level Distribution

Legitimate emails are classified as Low risk in 99.7% of cases and never reach High risk. This is operationally important as it confirms that High risk flags provide reliable indicators of phishing.

Human phishing achieves perfect High risk recall (100%), confirming that human-authored phishing attacks are highly sensitive to the type of urgent, threatening, and personally targeted language that PhishScore's social engineering features capture.

AI phishing shows the greatest class variation: 75.0% Medium risk and 25.0% Low risk, with none reaching High risk. This pattern reflects moderate social engineering keyword activation and greater stylistic complexity. The 25% Low-risk misclassification rate is consistent with Eze and Shamir's finding that standard detectors miss a substantial proportion of LLM-generated phishing attempts.

The distribution of PhishScore risk levels on the three email categories gives insight into the framework's effectiveness in identifying between legitimate email, human phishing and AI generated phishing emails.

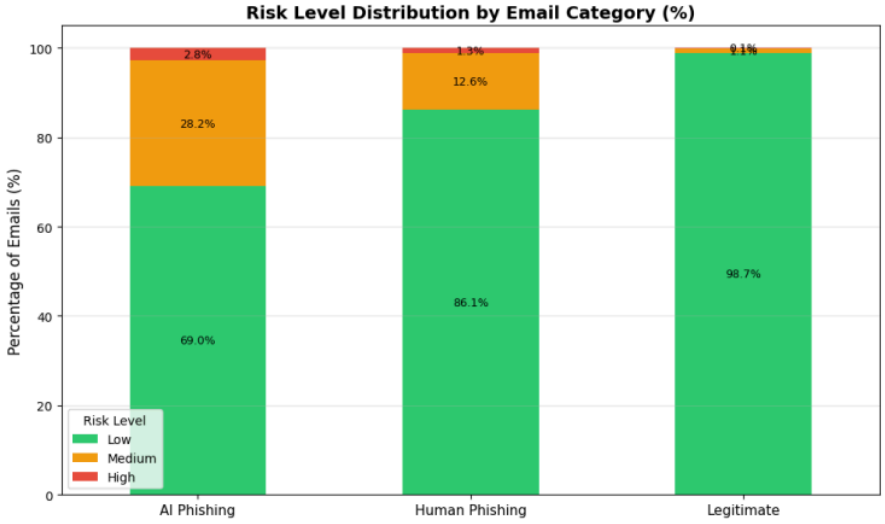

Figure 4.5: Stacked percentage bar chart of risk level distribution by email category

Overall, the distribution of risk levels shows that PhishScore is able to identify meaningful differences in behaviour among the three classes. AI-generated phishing falls somewhere between traditional phishing and the most sophisticated forms of spear-phishing, and the sophistication and ways in which AI phishing has escaped traditional detection methods illustrates the spread of the attack.In between the high levels of confidence traditional phishing is detected, and the more sophisticated forms of spear-phishing, lies AI-generated phishing, and the sophistication of AI phishing and how it is getting past traditional detection methods proves how it is spreading. These results confirm the importance of using behavioural and stylometric techniques to counter the growing threat posed by AI-powered cyber attacks.

4.5 Feature Profile Analysis

The legitimate class had negligible values in all social engineering features and word length average while a mean readability inversion score suggesting a standard but readable tone. The human phishing class features higher values for per sonal and threat scores than any other, and also demonstrates the basic phishing strategy of directly targeting the recipient and using dire consequences for inaction. Readability and word length are fairly normal as writers will attempt to write less complex emails than a machine may in order to maintain suspicion.

4.5.1 Heatmap of Normalised Feature Values

The heatmap reveals a distinct feature signature for each email type. Legitimate emails show consistently low social engineering and structural feature values and naturally higher lexical diversity and readability scores. Human phishing exhibits high urgency, threat, personal, and brand feature activations, validating its heavy reliance on psychological manipulation keywords. AI phishing shows the inverse pattern: high readability inversion

and average word length, indicating complex linguistic structure but moderate social engineering activation — confirming that AI phishing occupies a qualitatively different feature space. Figure 4.6 presents a heatmap of mean normalised feature values by email category.

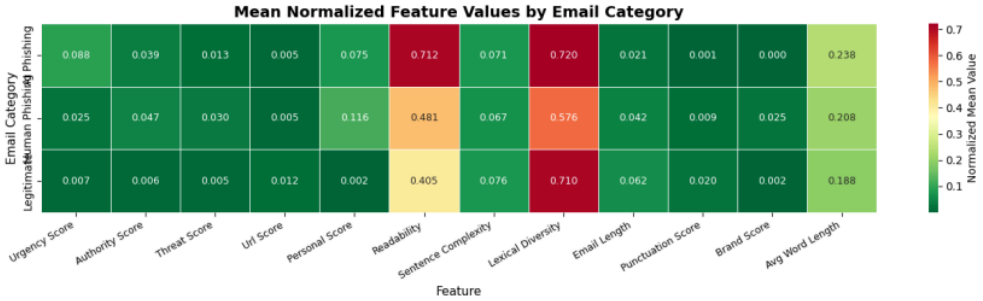

Figure 4.6: Heatmap of mean normalised feature values by email category

4.5.2 Radar Chart of Social Engineering Features

The radar chart reinforces the heatmap findings: human phishing (red) exhibits the largest and most asymmetric profile, with high personal and threat activations; AI phishing (purple) displays the smallest and most evenly distributed profile, reflecting moderate social engineering language; legitimate email (green) shows minimal keyword activation across all categories. Figure 4.7 presents a radar chart of mean raw social engineering keyword feature values per email category.

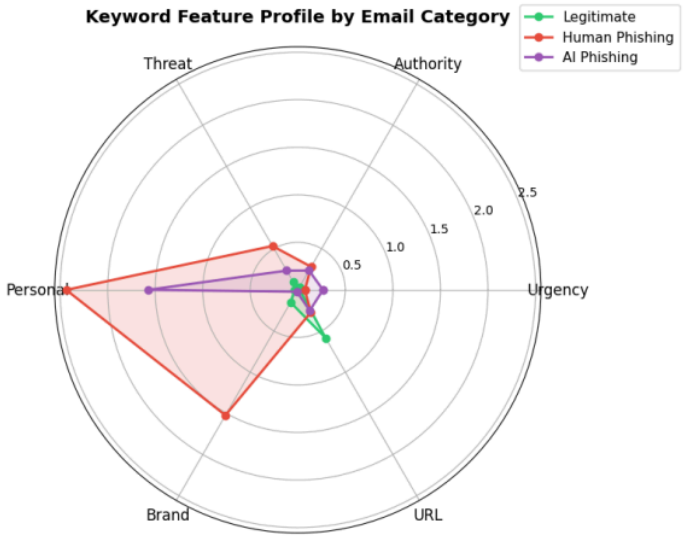

Figure 4.7: Radar chart of mean raw social engineering keyword feature values per email category.

4.6 Comparative Discussion

A qualitative comparison of PhishScore and four representative supervised phishing detection approaches given in the literature is presented in Table 4.1, where the five dimensions are compared: model type, training requirement, explainability, tri-class detection ability, and reported AUC or F1 performance. The most popular phishing baseline, which was the first to be widely used, integrates SVM with TF-IDF feature representations. This method provides competitive results with an F1 score of 0.92 on custom binary datasets, but requires a good amount of labeled data and has no interpretable output beyond a binary class label. Most importantly, it was built for human-made phishing that is not AI-generated, and this gap is becoming increasingly significant as phishing campaigns increase in number using LLMs to assist. It is also found to be sensitive to distributional shift across datasets, which is noted by Alhuzali et al., and a property that has been observed in its performance such that accuracy values obtained from one set of data are often not reliable in other sets of data.

Unlike other approaches, PhishScore doesn't require any training corpus, so it behaves the same way, regardless of the deployment environment. Specifically, fine-tuned BERT is the state-of-the-art algorithm for binary phishing classification with 0.96 F1 on several benchmark datasets. It lacks inherent explainability, however, and it is considered a binary classification problem, while fine-tuning and inference demands significant computational resources. For firms lacking access to large email datasets with annotations or to GPU hardware, practical deployment is still an obstacle.

The dual-path architecture suggested by Altan et al. combines the email body analysis approach based on transformer with structural analysis of URLs to complement each other, which further improves the reported F1 value to around 0.97. But, it is also a supervised training process, with only some levels of explainability and fails to cover a specific threat category: AI generated phishing. It also relies on the inclusion of raw URL strings in the text of emails, which is not always the case in body-only corpora, as confirmed by the analysis on url scores performed in Section 4.6 of this thesis.

Method	Model Type	Training	Explainable	Tri-class	AUC / F1	Dataset	Ref
SVM + TF-IDF	Classical ML	Yes	No	No	∼0.92 F1	Custom	[29]
BERT (fine-tuned)	Transformer	Yes	No	No	∼0.96 F1	Multiple	[24]
DistilBERT + SHAP	Transformer	Yes	Yes	Yes	∼0.93 F1	Enron+AI	[25]
Dual-path NLP+URL	Transformer+Struct.	Yes	Partial	No	∼0.97 F1	Custom	[26]
PhishScore (Ours)	Weighted Scoring	**No**	**Yes**	**Yes**	**0.9900 AUC**	Enron+Naz+AI	—

Table 4.1: Comparison of PhishScore with Representative Phishing Detection Approaches

The most similar prior art is the DistilBERT model and the SHAP-based explainability by Kumar et al. that works with the same data sources (Enron and AI phishing) that are utilized in this thesis, and that includes post-hoc feature importance scores for analysts to review. It attains around 0.93F1. This is a positive step in the direction of explainable tri-class detection, though SHAP explanations are created after the fact as approximations of how a model acts, and not necessarily a true reflection of a model's decision making process. By contrast, PhishScore is exactly explainable since its score is a linear sum over 12 interpretable features, where the numerical contribution of each feature to the final score is easily computed, without approximation. Furthermore, DistilBERT has to be trained with labeled data, and frequently needs to be retrained as phishing language changes, which PhishScore does not need to do because of its weight-based design.

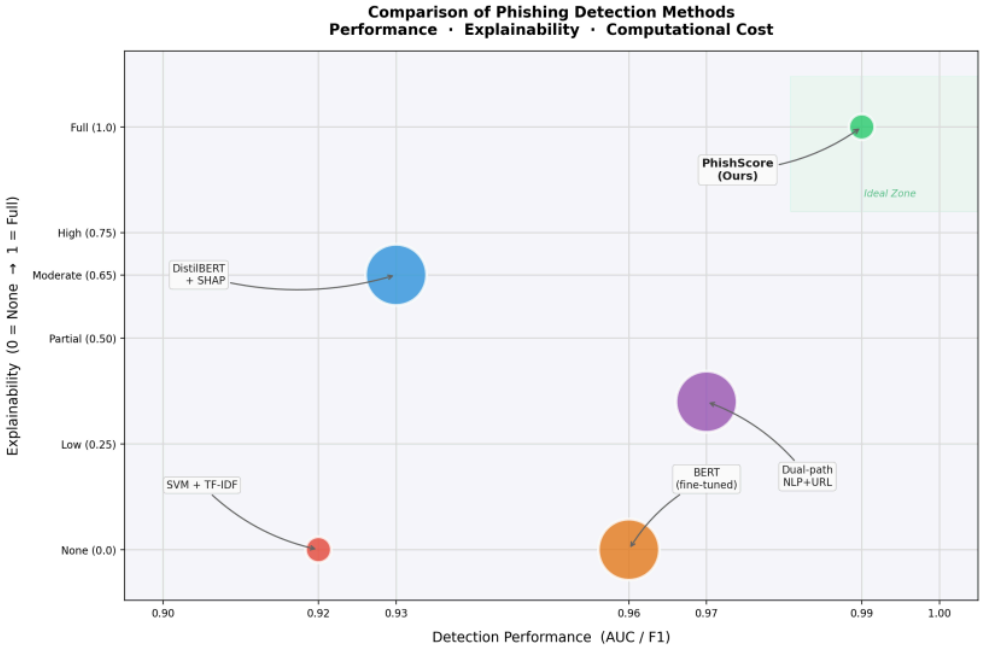

Figure 4.8: PhishScore vs. supervised methods — balancing detection performance, interpretability, and deployment cost

The sizes of these bubbles corresponds to the costs of computing the deployment of that method. The smallest bubbles represent the two systems with low deployment overhead: PhishScore (no training whatsoever) and SVM + TF-IDF (lightweight fitting on a labeled dataset). The three transformer-based methods, BERT, DistilBERT + SHAP, and the dual-path NLP+URL model, are shown as larger bubbles, which denote that they are more costly in terms of computational resources and require pre-training and fine-tuning on labeled corpora and GPUs to run inference. A smaller bubble is thus desirable, meaning that the method can be implemented using fewer resources, with less labelled data, and lower operational cost.

PhishScore is easily able to meet all three deployment requirements, and achieves an ROC-AUC of 0.9900, competitive with fully supervised models, while having complete feature-level interpretability and tri-class discrimination between legitimate, human-authored, and AI-generated phishing. Only one of these can be accomplished in this comparison. It is important to recognize, however, that direct comparisons of performance are necessarily constrained by variations in the type of data used to assess performance, the definition of classes and the experimental procedures used across the studies. It would require a controlled head-to-head evaluation on a common tri-class benchmark with equal preprocessing conditions for firm conclusions. A key point that the comparison does make is that a seemingly well-designed, training-free scoring model can provide competitive detection performance with a significant reduction in the operational overhead of supervised deployment; especially in resource-limited environments or in situations where phishing patterns are changing faster than labelled datasets can be collected.

Chapter 5

CONCLUSION AND FUTURE SCOPE

This dissertation addressed two complementary challenges in the modern cybersecurity landscape: the need for a comprehensive understanding of how AI-based behavioural threat detection has evolved, and the need for a practical, interpretable phishing detection framework capable of identifying AI-generated phishing — a class of threat for which existing binary supervised classifiers are fundamentally inadequate. The systematic literature review of eighteen studies traced a clear evolutionary trajectory in behavioural threat detection: from classical machine learning methods (SVM, Random Forest, Decision Trees) reliant on manual feature engineering, through deep learning approaches (CNN, LSTM, Autoencoders, GNNs) that enable automatic feature extraction but sacrifice interpretability, to LLM-based systems (SecurityBERT, APT-LLM, OMNISEC) that encode behavioural data as language and enable semantic reasoning, and finally to emerging agentic architectures that couple LLM reasoning with persistent memory and external tool invocation for autonomous, multi-step threat investigation. Each paradigm advance was shown to address specific limitations of its predecessor while introducing new challenges — most notably the interpretability deficit that pervades deep learning-based detection systems.

The PhishScore framework was developed in direct response to this interpretability deficit and to the specific challenge of AI-generated phishing detection. Key design choices — training-free operation, linear scoring architecture, twelve interpretable features spanning social engineering, structural, and stylometric dimensions, and fixed threshold risk classification — collectively produce a system that is transparent, deployable without labelled data, and computationally suitable for real-time enterprise SIEM integration. Empirical evaluation on 2,139 balanced emails demonstrated strong discriminative performance: ROC-AUC of 0.9900, ANOVA $F = 19{,}503.87$ ($p < 0.0001$), Cohen's $d > 2.9$ for all class pairs, and 100% High-risk recall for human phishing. The finding that average word length — a stylometric feature — is the strongest single discriminator of AI-generated phishing represents a counter-intuitive but empirically robust result: how an email is written is more diagnostically informative than what specific words it contains, at least for the purpose of distinguishing AI authorship.

Several directions for future research emerge from this work.

Integration into Multi-Agent Pipelines. Embedding PhishScore as a pre-filter at the first stage of a multi-agent security pipeline — routing Medium and High-risk emails to a reasoning agent for retrieval-augmented semantic analysis, threat intelligence correlation, and natural language explanation generation — represents the highest-impact near-term extension. PhishScore's linear time complexity makes it directly suitable for SIEM integration as a fast, interpretable first-pass filter before invoking computationally expensive LLM-based analysis.

Data-Driven Weight Optimisation. The current domain-knowledge-derived weights could be replaced with data-driven weights obtained through constrained regression or Bayesian optimisation on labelled data, increasing discriminative power while preserving the interpretability of the linear scoring structure.

Adaptive Threshold Calibration. The fixed thresholds at 33 and 66 could be adjusted by deployment context to reflect organisational security posture — a more security-conscious environment might lower the High-risk threshold, accepting higher false positive rates in exchange for reduced missed detection.

Multilingual Extension. The current keyword dictionaries and stylometric features are English-language specific. Extending PhishScore to non-English phishing campaigns — increasingly common as LLMs enable high-quality multilingual text generation — would substantially broaden real-world applicability.

Federated Weight Optimisation. Organisations may wish to share statistical information about attack patterns without sharing email content. A federated learning approach to weight optimisation would enable multiple organisations to collaboratively improve PhishScore weights while preserving data privacy.

The broader agentic paradigm identified in the systematic review suggests an independent research agenda encompassing autonomous vulnerability assessment, orchestration of incident response, and continuous threat hunting across diverse data streams — challenges that will require solving open problems in agent coordination, prompt injection resistance, and evaluation methodology. The same LLMs that enable stylometric detection are simultaneously lowering the barrier to more sophisticated phishing. This is an ongoing arms race, not a solved problem. This dissertation contributes a concrete, interpretable, and deployable step towards the agentic security future that this field is rapidly approaching.

Bibliography

[1] R. Abdallah et al., "Intrusion detection systems using supervised machine learning techniques: A survey," in *Proc. IEEE GLOBECOM*, 2021.

[2] M. Alsamir and M. AlShaher, "A comparative study of machine learning algorithms for network intrusion detection," *Int. J. Comput. Sci. Network Security*, vol. 21, 2021.

[3] S. Satpathy et al., "Random forest-based intrusion detection system," *Int. J. Innovative Technology and Exploring Engineering*, 2019.

[4] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 11, 2021.

[5] J. Hemalatha et al., "An efficient DenseNet-based deep learning model for malware detection," *Entropy*, 2021.

[6] A. Halbouni et al., "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, 2022.

[7] A. Idouglid et al., "LSTM-based network intrusion detection for adaptive cyber defense," *Electronics*, 2022.

[8] C. Dong and I. Kotenko, "GAN-based approaches for network intrusion detection," in *Proc. IEEE CyberSec*, 2021.

[9] C. Lee et al., "Graph neural network-based intrusion detection system," *IEEE Trans. Netw. Serv. Manag.*, 2021.

[10] S. Ullah et al., "A scheme for generating a dataset for anomalous activity detection in IoT networks," in *Proc. Canadian AI Conf.*, 2022.

[11] M. Keshk et al., "An explainability-based backdoor attacks detection approach in federated learning," *IEEE Trans. Artif. Intell.*, 2021.

[12] M. Hassanin and N. Moustafa, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[13] M. Rahman et al., "ChatGPT for cybersecurity: A systematic mapping study," *Computers & Security*, 2023.

[14] M. A. Ferrag et al., "SecurityBERT: A novel pre-trained language model for cybersecurity NLP," in *Proc. IEEE CSCloud*, 2023.

[15] S. Benabderrahmane et al., "APT-LLM: Embedding-based anomaly detection for cybersecurity using large language models," 2022.

[16] D. Palma et al., "LLM-based log analysis for cybersecurity," 2024.

[17] J. Cheng et al., "OMNISEC: LLM-driven threat intelligence for multi-stage intrusion detection," 2023.

[18] V. Vinay, "Agentic AI systems in cybersecurity: A generational perspective," 2024.

[19] Z. Ali and O. Ghanem, "LLM agents for autonomous cybersecurity operations," 2023.

[20] H. He et al., "Challenges and opportunities in deploying LLM-based security agents," 2024.

[21] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A systematic literature review on phishing email detection using natural language processing techniques," *IEEE Access*, vol. 10, pp. 65703–65727, 2022.

[22] M. Acharya and S. Sharma, "Detecting phishing emails through stylometric analysis: A novel NLP approach," in *Proc. IEEE ACROSET*, Indore, India, 2025, pp. 1–7.

[23] K. Przystalski, J. Argasiński, I. Grabska-Gradzinska, and J. Ochab, "Stylometry recognizes human and LLM-generated texts in short samples," 2024.

[24] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing phishing email detection: A comparative study of deep learning models," *Sensors*, vol. 24, no. 7, p. 2077, 2024.

[25] A. A. Kumar, T. P. Imthias Ahamed, and N. A. Shukoor, "Detecting human-written and AI-generated phishing emails with DistilBERT and explainable AI," in *Proc. IEEE InCoWoCo*, India, 2025, pp. 1–5.

[26] I. Altan, A. Bachir, Y. Parbhulkar, A. M. Rizvi, and M. Farazi, "Dual-path phishing detection: Integrating transformer-based NLP with structural URL analysis," 2025.

[27] C. S. Eze and L. Shamir, "Analysis and prevention of AI-based phishing email attacks," *Electronics*, vol. 13, no. 9, p. 1839, 2024.

[28] C. Opara, P. Modesti, and L. Golightly, "Evaluating spam filters and stylometric detection of AI-generated phishing emails," *Expert Systems with Applications*, vol. 276, p. 127044, 2025.

[29] A. Mittal et al., "Phishing detection using natural language processing and machine learning," vol. 6, 2022.

[30] A. Alhuzali, A. Alloqmani, M. Aljabri, and F. Alharbi, "In-depth analysis of phishing email detection: Evaluating the performance of machine learning and deep learning models across multiple datasets," *Applied Sciences*, vol. 15, p. 3396, 2025.

[31] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing email detection using natural language processing techniques: A literature survey," *Procedia Computer Science*, vol. 189, pp. 19–28, 2021.

[32] P. N. Wosah, Q. A. Mirza, and W. Sayers, "Analysing the email data using stylometric method and deep learning to mitigate phishing attack," *Int. J. Information Technology*, vol. 17, pp. 3823–3834, 2025.

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis_____

_____

Total Pages _____ Name of the Scholar_____

Supervisor (s)

(1)_____

(2)_____

(3)_____

Department_____

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: _____ Similarity Index: _____, Total Word Count: _____

Date: _____

Candidate's Signature **Signature of Supervisor(s)**

width=!,height=!,pages=-

width=!,height=!,pages=-

PHISHSCORE: A WEIGHTED MULTI-FEATURE SCORING FRAMEWORK FOR TRI-CLASS PHISHING EMAIL DETECTION

ORIGINALITY REPORT

9%	7%	4%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.dtu.ac.in:8080 Internet Source	2%
2	Submitted to Kurukshetra University Student Paper	1%
3	www.dspace.dtu.ac.in:8080 Internet Source	1%
4	Submitted to Delhi Technological University Student Paper	1%
5	T. Venkat Narayana Rao, Ananya Seeta, J. V. P. Udaya Deepika, Kumari G. Seshu. "chapter 18 Harnessing Deep Learning for Enhancing Security for Social Empowerment", IGI Global, 2025 Publication	<1%
6	www.mdpi.com Internet Source	<1%
7	Submitted to University of Wollongong Student Paper	<1%

8	"Artificial Intelligence Driven Forensics", Springer Science and Business Media LLC, 2026 Publication	<1 %
9	Xiao, wenjun. "MS-MambaTM: Multi-Scaled MAMBA Transformer Mixer for Medical Image Classification", The George Washington University, 2025 Publication	<1 %
10	Submitted to University of Birmingham Student Paper	<1 %
11	kclpure.kcl.ac.uk Internet Source	<1 %
12	Submitted to Universidad de Costa Rica Student Paper	<1 %
13	Submitted to Heriot-Watt University Student Paper	<1 %
14	Submitted to University of Nottingham Student Paper	<1 %
15	libres.uncg.edu Internet Source	<1 %
16	www.diva-portal.org Internet Source	<1 %
17	www.mathstat.dal.ca Internet Source	<1 %

18	Submitted to Universitas Islam Riau Student Paper	<1%
19	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
20	Kun-ta Chuang. "", IEEE Transactions on Knowledge and Data Engineering, 4/2007 Publication	<1%
21	Paolo Ferro, Harinadh Vemanaboina, Chander Prakash. "Computational Techniques and Smart Manufacturing", CRC Press, 2026 Publication	<1%
22	Submitted to Staffordshire University Student Paper	<1%
23	cyberleninka.org Internet Source	<1%
24	wp.cs.hku.hk Internet Source	<1%
25	www.acns.org.au Internet Source	<1%
26	Submitted to American Public University System Student Paper	<1%
27	L E Pablo. "Optic nerve head changes in early glaucoma: a comparison between	<1%

	stereophotography and Heidelberg retina tomography", Eye, 02/13/2009 Publication	
28	arizona.openrepository.com Internet Source	<1%
29	www.irjet.net Internet Source	<1%
30	Submitted to University of Essex Student Paper	<1%
31	link.springer.com Internet Source	<1%
32	www.thinkers360.com Internet Source	<1%
33	Submitted to National Institute of Business Management Sri Lanka Student Paper	<1%
34	Petra Perner. "An architecture for a CBR image segmentation system", Engineering Applications of Artificial Intelligence, 1999 Publication	<1%
35	eudoxuspress.com Internet Source	<1%
36	iieta.org Internet Source	<1%
	m.moam.info	

37	Internet Source	<1%
38	www.frontiersin.org Internet Source	<1%
39	cin.philab.esa.int Internet Source	<1%
40	docserv.uni-duesseldorf.de Internet Source	<1%
41	scholarcommons.usf.edu Internet Source	<1%
42	theses.hal.science Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		