Systematic Literature Review on the Semantic Analysis of Cybersecurity Indicators

Nic Recasens
Computer Science
Georgia Southern University
Statesboro, Georgia, USA
jr38088@georgiasouthern.edu

Nigel Smith
Computer Science
Georgia Southern University
Statesboro, Georgia, USA
ns15468@georgiasouthern.edu

I. Introduction

In the modern world, attackers are a step ahead of all Cybersecurity professionals. Professionals need to be able to adapt to every threat out there, while attackers only need to be able to adapt to a single fatal weakness. As such, the realm of machine learning and predictive analytics is of known importance in Cybersecurity research. For decades, machine learning analysis has helped keep defenders on top of their game by preventing novel attacks from gaining footholds. In the 90s, when the number of attackers was smaller, professionals were able to get away with only blocking known threats. But with Ransomware and Malware becoming billion dollar multi-national black markets, we need to be able to apply risk trends to novel data. At the forefront of this prevention is the identification and blocking of spam and phishing. An individual datum related to spam and phishing is called an indicator of attack (IoA), and they let us know if someone is targeted by an attacker. And if they are used successfully, they become indicators of compromise (IoC). IoCs have the negative reputation of being indicative of failed Cybersecurity protection, but in relevance to this project, they are the second half of the puzzle that is semantic analysis of indicators. Via the correlation of IoAs and/or IoCs, numerous researchers have pushed the boundary forward over the decades since the aforementioned 1990s. The attackers may be numerous in their successes, but they leave behind their own value as well: data. With that data, we can see how pervasive cyber threats are, train models to simulate said attackers, and allow the very computers they attack help mitigate the risks themselves. Through this literature review, we will go over 8 papers that show how researchers achieved these goals, and show through their research how important cyber-risk mitigation truly is in our modern interconnected world.

II. Methodology

At the most basic level, the 3 keywords we used to guide our literature selection process were "spam", "phishing", and "semantic". Spam and Phishing are the strongest indicators Cybersecurity professionals have as indicators of attack. In fact, Business Email Compromise or (BEC) can be identified as the largest threat to the modern digital business [5]. As such, once we have papers scoped to the huge risk spam and phishing compose, the "semantic" keyword allows us to predominantly find Computer Science papers detailing how semantic analysis can help with this cause. Some linguistic journals do appear with these keywords, but with the current rise of the Large-Language Model, semantic research is seeing a heavy Computer Science edge.

For searching these terms, Google Scholar was the primary resource, though GALILEO also was utilized for accessing some non-free files. When on our Georgia Southern University network, some research journals allow our IP to access full papers without logon, which was also useful for reading full texts. Publication Year and citation information was considered and prioritized past the above keyword filtering, and was based on the Google Scholar data for those indicators.

And to wrap up our selection process, our primary exclusion criteria was mentioned earlier, and that is a Computer Science focus. Some articles were industry paid, so were more like Sociology in nature. And for other articles, they were focusing on the behavioral aspects of phishing and spam, which while useful, are not in scope for this review. And speaking of scope, the primary *inclusion* criteria was related to this course's guidelines, and that is publication year. Many useful articles at the beginning of semantic analysis for spam prevention were made in the late 2000's, but they were specifically asked to be ignored for this paper by the paper requirements. As such, only 'recent' research was provided, meaning publication dates past 2020 as much as it could be helped. These criteria, alongside with the above requirements and toolset mindsets, allowed us to choose the below 8 papers to systematically review.

III. Literature Review

A. 2019: iMCircle: Automatic Mining of Indicators of Compromise from the Web

To begin, we start with our oldest paper from 2019, as this paper shows the limitations of the research into spam indicators before the usage of modern semantic ML techniques. Authors Zhang, Ya, Liu, Li, Shi, and Gu for this paper prioritized the fact that before we can remediate and prevent spam and phishing, we have to know what spam and phishing actually *is* via data collection. And while that is a valorous cause for research, there are significant limitations with traditional Cyber Threat Intelligence (CTI) gathering regarding those indicators.

Modern CTI often relies on a "fixed-point" monitoring of curated data feeds from sources like security organizations and white papers [7]. It also should be noted that "fixed-point" here refers to when detection systems make a baseline "secure" configuration, and an institution then marks all modifications to that baseline as insecure. This conventional approach is described as time-consuming and heavily complicated by the authors because it required researchers to clean data from multiple disparate sources, and to wait for information to be released by data providers as public. Consequently, current methods may suffer from incomplete coverage and data delays, meaning these traditional methods fail to keep pace with the rapidly evolving landscape of cyber threats, as every second counts when triaging.

To solve this, the authors developed a system called iMCircle. The iMCircle system employs an active recursive architecture consisting of four major components: a search engine crawler, an information preprocessor, an IoC checker and generator, and a new "seed" extractor (with seeds being the inputs for the next run of the iMCircle recursion) [7]. Unlike former passive systems, iMCircle takes spam and phishing indicators (like domains and IoCs) and combines them with a "target threat field" (i.e. "APT attack") to create search engine queries for exploratory data collection. It uses Google to retrieve relevant public information, which is then processed using Natural Language Processing (NLP) techniques to extract semantic features from site titles, URLs, and page snippets [7]. Machine learning models like Support Vector Machines (SVM) and Decision Trees then classify whether an indicator is a legitimate threat. In the final step of the chain, Validated IoCs are then used to extract new seeds from their associated webpages, restarting the cycle.

The study's primary contribution with this cyclical approach is the development of the first system (according to the authors) that mines IoCs continuously and automatically by checking suspicious indicators against open-source threat information. Since the time of this paper as well, we have seen many security scraping efforts be prioritized as well [7]. During a two-month real-world evaluation, the system demonstrated significant efficacy: the SVM model achieved a precision of 96.70% and an F1 score of 90.46% in identifying threat indicators. Furthermore, iMCircle successfully discovered 2,941 unique data sources, including more niche and varied public sources that traditional monitoring might overlook. That ability to get fresh data, rather than recycled, was of considerable satisfaction for the authors [7].

But, while iMCircle showed impressive performance, the paper did still acknowledge several limitations. Firstly, the system is entirely dependent on open-source information available on the Web; it cannot identify or verify suspicious indicators that have not been publicly discussed or indexed by search engines. This also means it was not tested on private repository information either for organization specific results [7]. There is also a disproportionate reliance on the quality and accuracy of search engine results, with the authors noting that crawling restrictions and empty results can reduce and fluctuate the usable sample size [7]. And lastly, the current implementation primarily focuses on a few metrics like domain names and Advanced Persistent Threat attacks (APT). So while the authors suggest that iMCircle could be adapted for other indicators like IP addresses or botnets, novel combinations like those would require further validation to prove the system's comprehensiveness across the entire threat landscape.

B. 2020: A semantic-based classification approach for an enhanced spam detection

The 2020's were a boom in semantic research, as we saw more powerful AI models hit impressive feats. As such, the research of Saidani, Adi and Allili in the beginning of this year addresses how the persistent and evolving threat of email spam can be mitigated by incorporating these new paradigms. Spam continues to account for over 56% of total email traffic, but traditional spam detection systems largely rely on the provenly ineffectual "bag-of-words" model, which evaluates emails as an unstructured collection of independent words without considering grammar, context, or word order. Semantic analysis can help prevent one of the biggest risks of BoW that the authors identified, which is polysemy [5]. Polysemy is a linguistic problem where a single word like "bow" can have drastically different meanings depending on the context. Spammers purposefully exploited the weaknesses of basic keyword filters through obfuscation, leading to the semantic classification employed in the paper.

To combat these tactics, the authors proposed a novel two-level semantic analysis framework. In the first level, incoming emails are categorized into distinct subject domains (such as Health, Finance, Education, Computer, and Adult). This establishes a tailored dataset of spam indicators for each specific sector [5]. In the second level, the researchers build a "feature space" for each domain by combining regular expression formatted hyperparameters with automatically extracted rules generated via the CN2-SD top-down search algorithm [5]. These rules are then merged and applied against test messages, with various machine learning algorithms being tested. These included Adaboost, Random Forest, and Naïve Bayes, with the goal to construct classifiers that accurately differentiate between spam and legitimate email indicators semantically.

Moving past lexical analysis to semantics led to the central finding of the study, which was that the domain-specific semantic approach significantly outperforms general-purpose classification models, including enhanced topic-based vector space models (eTVSM) (which were the high standard before this paper). Some domains were able to be trained successfully enough to reach 98% accuracy [5]. A major concept that this research contributed is proving that while spammers can always manipulate spelling and utilize synonyms to trick BoW classification, they cannot easily disguise the underlying intent or meaning of their emails (as of when this paper was published) [5].

Now, despite the high accuracy in multiple domains, the proposed semantic framework has areas requiring further

development. The authors acknowledge that achieving "word sense disambiguation" could clearly improve their work, with word sense referring to the ability of models to disambiguate context around a phrase. Word sense is an ongoing challenge, and to the authors represents the next frontier for future work in spam detection [5]. While domain categorization narrows down the context, teaching a computer system to definitively resolve nuanced word meanings based purely on complex sentence structures is still difficult. Additionally, the reliance on manually specified rules as part of the hybrid approach means that experts must continuously update their knowledge engineering to maintain efficiency and keep pace with the rapidly evolving tactics of modern spammers [5].

C. 2020: Detecting Phishing SMS Based on Multiple Correlation Algorithms

Now that we've started getting the foundations of this research topic laid, this next paper from Sonowal and Gunikhan shows how breadth of study for semantic spam blocking entails so much more than just emails. When we discussed spam indicators earlier, email shows up as the most common vector, but these authors have a notable paper showing how cellular communications also hurt end users. This is introduced as "smishing", or SMS phishing. Importantly, it is characterized as a disproportionately dangerous form of cyberattack due to the high trust users place in text messaging. Unlike traditional email delivery methods, which see open rates of roughly 20-30%, SMS messages boast a staggering 98% open rate, making them an ideal delivery system for malicious and unwanted messages [6]. Traditional defense mechanisms like static blacklists and keyword filters are weakened by the modern attacker trends of URL shorteners, swapping similar-looking characters like O and 0, and social engineering through falsely promoting urgency and legitimacy. And to top off the challenges SMS face, the study also identifies a central technical hurdle: high-accuracy detection is difficult to scope when we are talking about a mobile device. A desktop computer has a powerful full-sized CPU, and is chained to a wall. But a phone has much more stringent power, heat, and battery constraints when the topic of machine learning is concerned [6].

To combat the aforementioned newer tactics, the researchers went away from the older "bag of words" keywords, and instead used a more dynamic "features" paradigm to profile text. They identified 52 distinct features, including textual metadata like message length and capitalization ratios, and more complex linguistic metrics like the Gunning Fog and Coleman-Liau Indices [6]. A notable metric the authors uncovered was the dichotomy between standard business email template and the erratic failed syntax of spammers, with it allowing high consistency in detection.

To process this data, they utilized the AdaBoost (Adaptive Boosting) classifier, an iterative ML model that prioritizes learning from its own classification errors to distinguish between the differences in true and false positives. And as for the ranking algorithm, Kendall rank correlation provided superior accuracy over other correlation algorithms in prioritizing categorized results. These together helped lower the 52 features from earlier down to just 20 . That is a 61.53% decrease in dimensions, all while still achieving an impressive 98% accuracy [6]. That reduction in dimensionality was key in making the model viable for mobile devices' lower power draw.

But, despite all the successes in the SMS field the authors enjoyed, there is still work to be done in non-email based instances. First, as generative AI tools become more indistinguishable when mimicking the professional readability scores of companies, the "readability dissonance" the researchers rely on may diminish [6]. Additionally, while the paper focuses on the positives of tracking features, in terms of user privacy, the surveillance concerns of an AI analyzing a user could limit further adoption of the author's security measures.

D. 2022: An Effective Cybersecurity Awareness Training Model: First Defense of an Organizational Security Strategy

This moves us into the year 2022, and here, with the effect of spam and phishing fully realized in the previous papers, there should be no surprise that a continued effort exists for preventing attacks from successfully landing. For the majority of the papers reviewed, we have seen how computers can protect us from threats. However, when it comes to protecting ourselves, protecting the human factor, the work of Dash and Ansari shows how computers can help us improve proactively, rather than rely on content filtering alone. For false negatives, user training is a key part of security, and in a way, can be seen as teaching humans the ways to spot the indicators ML saw earlier. Despite advancements in defensive systems like firewalls and email filtering, "naïve information security practices" as the authors put it accounts for between 72% and 95% of all cyber threats [2].

To address these vulnerabilities, the study employs a novel qualitative research design method utilizing the Technology Acceptance Model (TAM) as its primary theoretical framework. TAM was created in 1989 to aid human-computer collaboration, but the authors add modifications that incorporate "behavioral intention" to relate it to modern training needs [2]. This helps us to understand how intervention through training can successfully influence employee behavior. Additionally, the paper evaluates the National Initiative for Cybersecurity Education (NICE) and the viCyber model. The viCyber model is a cloud-based AI system that uses visual mapping and an inference engine to develop a customized cybersecurity curriculum based on specific competency needs, while NICE is the framework the approach relies on [2].

The study found in order to be effective, security strategy must adopt a bottom-up approach. This means making every employee accountable for the security of their own representative areas, rather than blaming software vendors or IT [2]. A significant contribution of this research past that is the endorsement of AI-enabled training tools like viCyber. AI tools like this can provide real-time feedback, and can even adapt to each user's understanding level, allowing the

aforementioned accountability to come more naturally [2]. The authors found that adaptable systems like these are more effective than traditional "one-size-fits-all" Massive Online Open Courses (MOOCs) because they can be tailored in efficient and individualized ways [2].

But, the paper still did not completely solve the user education crisis. While they do advocate for robust personalized security training to protect users from phishing, the authors also identified a literature gap regarding AI-based security training efficacy, which the study itself barely begins to fill. Another notable limitation here is the warning that high expectations from executives can create an unhealthy security climate if technical defense is neglected. Just because better training supports security teams doesn't mean that other security factors can just be let go [2]. Additionally, the paper mentions that the viCyber tool is still a "work in progress", and acknowledges that many organizations struggle to implement complex frameworks like NICE due to a lack of domain experts on payroll. This all indicates that while the concepts are indeed sound, they are just expectedly not a silver bullet without further development.

E. 2023: Phishing or Not Phishing? A Survey on the Detection of Phishing Websites

Looking to existing survey literature: Zieni, Massari, and Calzarossa provide a nearly comprehensive taxonomy of phishing website detection methods available to end users. The core problem addressed by this survey is that there is not a single technical gap, but rather a fragmentation of the detection landscape itself. The Anti-Phishing Working Group (APWG) reported over one million unique phishing websites observed just in the first quarter of 2022 [8]. Despite the size of this prevailing internet poison, there is no unified framework for comparing detection approaches across their respective strengths and weaknesses. Prior surveys had either focused on a single detection paradigm, such as visual similarity or ML alone, or had limited their scope to papers published before the recent surge in deep learning applications [8], that is to say, the research field of utilizing ML/AI tools for phishing detection is still in its infancy, as with all subfields of cybersecurity and software engineering. To address this, the authors conducted a structured literature review beginning with a broad Scopus query for the term "phishing," which returned $\tilde{4}$,400 papers, narrowed to 600, and then further refined to 127 papers selected by venue quality, citation count, and community interest [8]. From these papers, they organize the detection literature into three principal categories: list-based methods, similarity-based methods, and ML-based methods. List-based approaches, which rely on blacklists and whitelists of known URLs, are simple and fast but fundamentally reactive. Because these lists can only be updated after an attack has been identified, they are inherently unable to defend against zero-hour attacks, where a brand-new phishing URL has not yet been flagged [8]. Several papers in the survey propose methods for proactively populating blacklists by predicting new phishing URLs from previously flagged ones, but the authors note that even these predictive heuristics struggle with the low cost attackers face in generating fresh URLs. Similarity-based approaches, such as TF-IDF scoring, Earth Mover's Distance for image signatures, and SIFT-based keypoint matching [8], take a different angle by comparing suspicious web pages to their legitimate counterparts using textual content such as HTML, CSS, and DOM trees, or visual content such as page and logo snapshots. The authors find that while these methods can detect zero-hour attacks, they suffer from high computational and storage costs, dependence on subjective similarity thresholds, and vulnerability to evasion tactics like HTML obfuscation and replacing text with images [8]. The bulk of the survey is dedicated to ML-based methods, reflecting the dominance of this paradigm in the literature. The authors present a detailed feature extraction framework organized around what properties are useful for detection, where those properties are derived from, and how they are encoded into features [8]. URL-based features are subdivided into lexical, statistical, network-based, and reputation properties, while HTML-based features cover textual, visual, and traffic properties. The survey identifies Support Vector Machine and Random Forest as the two most commonly applied traditional algorithms, with Random Forest tending to outperform others across varied settings [8]. Deep learning models, particularly Convolutional Neural Networks and Long Short-Term Memory networks, are noted as an increasingly popular direction that avoids manual feature engineering but requires substantially more computational resources [8]. A key contribution of the survey is its comparative analysis across all three detection paradigms. The authors synthesize a clear set of recommendations: features should be chosen with attention to their discriminating power and vulnerability to attacker manipulation, datasets should be diverse and balanced, and model parameters should be disclosed to support reproducibility [8]. They also identify a huge research gap: the challenge posed by URL shortening services that render most lexical URL features meaningless. [8]. However, the survey itself is not without limitations. Because the literature search was anchored to Scopus with a cutoff around 2021, very recent developments in large language model-based detection and transformer architectures are not covered. This brings us to our earlier point of strong consideration: the field of using ML tools for phishing detection is still in its infancy. Additionally, while the three-category taxonomy is useful for organizing the field, hybrid approaches that combine elements of all three paradigms receive comparatively less attention.

F. 2025: Phish Fighter: Self Updating Machine Learning Shield Against Phishing Kits Based on HTML Code Analysis

Building on the detection taxonomy established by the preceding survey, Brezeanu, Archip, and Artene narrow their focus to a specific vector: phishing kits, reusable HTML templates that allow attackers to rapidly deploy fraudulent web pages with minimal modification [1]. The core problem is twofold. First, existing detection methods, whether list-based, visual similarity, or content analysis, struggle with zero-day

phishing pages that exploit previously unknown templates. Second, most phishing kit detection systems require prior knowledge of the kits in use and depend on manual or semi-automatic updates to remain current, meaning they degrade quickly as attackers iterate on their tooling [1]. The authors also observe that phishing kits frequently reuse identical HTML structures across campaigns targeting entirely different brands, with only superficial changes to images, text, and CSS styling, making the underlying code structure a more reliable detection signal than visual or textual content [1].

To address these gaps, the authors propose Phish Fighter, a three-module system. The first module, the Feature Extractor, processes the virtual DOM tree of a web page and produces a representative tag sequence by systematically removing noise: the head section, hidden elements, filtered tags like images and formatting, script tags, and redundant container tags such as nested divs and spans that attackers insert to disrupt fingerprint-based detection [1]. The module also normalizes synonymous HTML tags, for example treating ordered and unordered lists as equivalent, and preserves the id, name, and type attributes of input tags as distinguishing features of specific kits. This entire preprocessing runs in linear time relative to the number of HTML tags in the body section [1]. The second module applies AGNES agglomerative hierarchical clustering to the tokenized tag sequences using a weighted cosine similarity metric. Rather than requiring a predefined number of clusters, the algorithm iteratively increases its inconsistency threshold and selects the clustering that maximizes the silhouette score, dynamically determining the number of phishing kits present in the dataset [1]. Clusters with fewer than three members are labeled as uncertain rather than discarded, preserving them as potential seeds for future kit identification. The third module is a Random Forest classifier trained on the cluster labels. When a new web page is classified as uncertain, the system automatically triggers a re-clustering of the full dataset including the new page, and if a valid cluster forms, the classifier is retrained without any human intervention [1].

The system was evaluated using phishing pages collected from the Phishunt repository alongside phishing kits sourced from GitHub bundles including Zphisher, Shark, and BlackEye [1]. The classification results showed weighted precision, recall, and F1 scores all exceeding 90%, with micro-averaged versions of each metric surpassing 95%. The macro-averaged scores were lower, around 85%, which may be due to the severe class imbalance inherent in phishing kit datasets, where some kits appear hundreds of times and others only a few [1]. The authors demonstrate several practical strengths through case studies: the system correctly identified the same kit being used across brands like Facebook and DHL, detected kits reused across different languages, and successfully flagged phishing pages that were direct clones of legitimate websites like Amazon. The continuous update module proved capable of recognizing a new kit after encountering just three pages sharing the same template [1]. But again, we see the common thread: this does not result in zero-day, first sighting detection.

The authors do acknowledge limitations. The clone detection capability can backfire: if the system is initially trained on cloned pages without the legitimate original in the dataset, it could flag the real website as malicious [1]. They suggest URL analysis as a pre- or post-processing step to limit this, though they concede that URL-based methods carry their own weaknesses. Additionally, the system focuses exclusively on HTML structure and does not incorporate CSS, JavaScript behavior, or network-level indicators, which means it would not detect phishing attacks that rely on those vectors rather than on reusable HTML templates. Here we see a significant opportunity to further the existing research: CSS and JavaScript are fundamental in web usage in the current day. The dataset, while drawn from real-world sources, is also limited to the brands and kits available through Phishunt and the selected GitHub repositories, leaving open the question of how well the approach generalizes to entirely novel phishing ecosystems [1].

G. 2025: APOLLO: A GPT-based tool to detect phishing emails and generate explanations that warn users

Where the previous papers have addressed phishing detection through traditional ML classifiers and feature engineering, Desolda, Greco, and Viganò use a large language model as both a detection engine and an explanation generator. The central problem their work addresses is twofold. First, despite continued improvements in automated phishing detection, no existing system achieves perfect accuracy, meaning that borderline cases are inevitably passed along to the user in the form of warning dialogs [4]. Second, the warnings that users actually see in browsers and email clients are found to be widely ineffective. The authors trace this failure to several factors: passive warnings that users simply do not notice, the habituation effect where repeated exposure to the same generic message reduces attentiveness, and a lack of specific explanations about what makes a particular email dangerous [4]. Prior work had shown that adding explanations to warnings could improve user decision-making, but producing those explanations was a time-consuming process that could not easily scale to new attacks [4]. Our persistent theme reappears: zero-day detection and application of this tool is not a possibility. To address both the detection and explanation problems simultaneously, the authors developed APOLLO, a Python-based tool powered by OpenAI's GPT-4o. The system's architecture consists of three modules that operate in sequence. The preprocessor module takes a raw email file, strips its HTML, extracts the subject, headers, and body, and replaces embedded URLs, phone numbers, and email addresses with meta-tags to reduce the input length for the model [4]. The URL enricher module then queries two external services: VirusTotal, which scans URLs against over 90 antivirus detectors and blocklists, and BigDataCloud, which provides geolocation data for the domain's hosting server [4]. This external data enrichment is a form of retrieval-augmented generation, allowing the model to reason about factual threat intelligence rather than relying solely on its training data.

Finally, the LLM prompter module feeds the preprocessed email and URL intelligence into GPT-4o through a two-stage prompt chain. The first prompt asks the model to classify the email as phishing or legitimate, estimate a probability, identify any persuasion principles the attacker may have employed, and list three to five explainable phishing indicators. The second prompt takes the most relevant indicator from the first response and generates a brief, user-facing explanation structured as a feature description, a hazard explanation, and a statement of consequences, following established warning design guidelines [4]. The classification performance of APOLLO was evaluated on a dataset of 4,000 emails, evenly split between phishing and legitimate samples drawn from the Nazario, NigerianFraud, and SpamAssassin datasets [4]. Without any external URL data, GPT-4o achieved 97.4% accuracy, 0.964 precision, and 0.985 recall on the binary classification task. When the VirusTotal data was sufficiently mature, meaning at least 25% of detectors had rendered a verdict, accuracy rose to 99.9% with only four misclassifications out of 4,000 emails [4]. However, the evaluation also revealed an important caveat: when VirusTotal returned entirely uncertain data, as would be the case in the first moments of a zero-hour attack, the additional information actually hurt performance relative to using no URL data at all. Conversely, when deliberately erroneous VirusTotal data was injected, accuracy collapsed to below 45%, though the authors noted that GPT-4o's recall remained relatively robust, meaning it was far more likely to produce false positives than to miss actual phishing emails [4]. Repeated trials confirmed that the model's classification labels were stable across runs, though the confidence probabilities showed some variability. Beyond detection, the authors conducted an exploratory user study with 20 participants recruited through Prolific to evaluate how users perceived the LLM-generated warning explanations. Participants were exposed to four phishing emails, each paired with a warning containing an APOLLO-generated explanation, and their responses were compared against four baselines: a manually crafted warning with expert-written explanations, and the standard warnings from Chrome, Firefox, and Edge [4]. The results showed that APOLLO's warnings scored higher on understandability and interest than all four baselines, with medium to high effect sizes against the manual explanation and low effect sizes against the browser warnings. On trust, APOLLO's warnings outperformed Chrome and the manual baseline for multiple conditions. Notably, 76.2% of participants exposed to APOLLO's warnings chose the safest action of not continuing to the website, compared to only 52.8% for the manual explanation baseline [4]. The study also included a false positive condition in which a legitimate Facebook email was flagged with a forced explanation, and participants showed appropriately lower trust in that warning, suggesting the explanations helped users make more nuanced judgments rather than blindly following every alert [4]. The study does carry several limitations that the authors acknowledge. The user study was exploratory with only 20 participants, limiting generalizability. The quality of the explanation depends inherently on the prompt design, and the features chosen for the explanation were constrained to four specific indicators rather than being generated freely by the model [4]. The evaluation dataset, while large for classification purposes, contained emails over a decade old. Additionally, the system currently processes only the first URL in an email, and the individual contributions of VirusTotal versus BigDataCloud to classification improvement were not isolated. The authors also did not evaluate the system against other LLMs, though they note that a mixture-of-agents approach could potentially yield even stronger results [4]. The research done in this paper lends itself well to future modularity: applying these same methodologies using different models and/or dataset could be a worthwhile endeavor to improve the final product.

H. 2023: Which phish is captured in the net? Understanding phishing susceptibility and individual differences

Having now surveyed the technical landscape of automated detection means for phishing, we look back to Sarno, Harris, and Black for the human side of the equation. Their work addresses a fundamental problem that no amount of automated filtering can fully resolve: some phishing emails will inevitably reach the inbox, and when they do, the user becomes the last line of defense [3]. The specific challenge motivating this study is that prior research on the psychological predictors of phishing susceptibility has produced numerous conflicting findings. For example, impulsivity has been shown to increase susceptibility in some studies but decrease it in others, and the Big Five personality traits have yielded contradictory results across multiple experiments [3]. The authors attribute these inconsistencies to methodological limitations in the existing literature, including small or non-diverse email sets, reliance on retrospective self-report measures of susceptibility, and convenience samples drawn from single university populations [3]. To produce more robust evidence, the authors recruited 150 participants from MTurk, deliberately sampling equally across three age groups: younger adults aged 18 to 29, middle-aged adults aged 30 to 54, and older adults aged 55 and above [3]. Participants completed a battery of validated psychological instruments measuring the Big Five personality traits, five facets of impulsivity via the Short UPPS-P scale, three dimensions of curiosity from the Five-Dimensional Curiosity Scale, and a novel 14-item Phishing Awareness Questionnaire designed to holistically capture phishing knowledge, experience, and habits [3]. The email classification task itself used 50 real emails, evenly split between legitimate and phishing, Each phishing email was matched to a legitimate counterpart from the same brand, reducing the influence of content preferences on classification accuracy [3]. Participants classified each email, indicated what action they would take, and reported their confidence. [3]. Three separate regression models were constructed to predict email discrimination ability. The age model explained 10.4% of the variance and found that classification ability improved across the lifespan, with younger adults being the most susceptible, directly contradicting the popular assumption that older adults are the most vulnerable

population [3]. The personality model explained 42.6% of the variance, with extraversion negatively predicting discrimination ability and both agreeableness and openness to experience positively predicting it. Conscientiousness and neuroticism did not significantly contribute to the model [3]. The deficient self-regulation model, which captured impulsivity and response times, explained 32.2% of the variance. Higher impulsivity scores and faster classification times both predicted poorer discrimination, consistent with the Suspicion, Cognition, and Automaticity Model, which posits that impulsive email processing prevents users from becoming suspicious of phishing attempts [3]. A combined hierarchical regression incorporating all significant predictors explained 48% of the variance in discrimination ability. Notably, when personality and self-regulation variables were entered into the model alongside age, age was no longer a significant predictor, suggesting that the age effect observed in the simpler model may be driven by age-related differences in personality traits and self-regulatory tendencies rather than by age itself [3]. The correlation analyses also revealed a telling dissociation between self-perceived and actual ability: scores on the Phishing Awareness Questionnaire were unrelated to classification performance but were positively correlated with confidence, indicating that self-reported phishing experience may foster overconfidence rather than genuine detection skill [3]. The study has several limitations the authors acknowledge. The sample, while more diverse than many prior studies, was still drawn from a single crowdsourcing platform, and broader recruitment across socioeconomic and educational backgrounds could yield different patterns. The Phishing Awareness Questionnaire, though novel and internally reliable, failed to predict performance, and the authors suggest that broader constructs like digital literacy may be more sensitive predictors [3]. Additionally, the email set, while diverse and validated, consisted of untargeted phishing emails, leaving open the question of how individual differences interact with more personalized spear-phishing attacks. [3].

IV. Synthesis and Conclusion

Two common themes reveal themselves across these papers. First, existing tools are commonly vulnerable to zero-day attacks, which suggest the need for robust ML and deep learning tools. From the reactive blocklists reviewed by Zieni et al.\cite{zieni_phishing_2023} to the phishing kit clustering of Brezeanu et al.\cite{brezeanu_phish_2025} to the retrieval-augmented classification of APOLLO\cite{desolda_apollo_2025}, every system reviewed encounters the same fundamental constraint: detection accuracy degrades sharply when the threat has not been seen before. Brezeanu's Phish Fighter requires a minimum of three pages from a new kit before it can form a cluster, and APOLLO's performance actually worsens when VirusTotal returns uncertain data on newly minted URLs. Even iMCircle, which was designed specifically for continuous discovery, can only surface indicators that have already been publicly discussed somewhere on the open web\cite{zhang1}. The zero-day problem is not merely a limitation noted in passing by these authors; it is the central unsolved challenge that connects the entire body of work reviewed here.

Second, the ability to meet this need exists, but is still in its infancy, with effective ML tools having only existed for a matter of a few years. The progression visible across the papers is striking: Saidani et al.\cite{saidani1} advanced past bag-of-words with domain-specific semantic rules in 2020, Sonowal and Gunikhan\cite{sonowal1} showed that careful feature engineering could make ML viable even on resource-constrained mobile devices, and by 2025, APOLLO demonstrated that a general-purpose LLM could classify phishing emails at 97.4% accuracy without any hand-crafted features at all\cite{desolda_apollo_2025}.

Our project would aim to reconcile the findings and shortcomings of these papers by utilizing cutting-edge ML tools for the purpose of highly accurate and immediate phishing detection. Specifically, the research gaps identified here point toward a system that can operate effectively against novel threats without requiring a warm-up period of previously seen examples, that can leverage the semantic reasoning capabilities demonstrated by LLM-based approaches while maintaining the structural analysis strengths shown by methods like Phish Fighter, and that produces outputs interpretable enough to support the human end user who remains, as this review has shown, the last and most variable line of defense.

References

[1] Gabriela Brezeanu, Alexandru Archip, and Codrut-Georgian Artene. Phish fighter: Self updating machine learning shield against phishing kits based on HTML code analysis. 13:4460–4486.

[2] Bibhu Dash and Meraj Farheen Ansari. An effective cybersecurity awareness training model: First defense of an organizational security strategy. *International Research Journal of Engineering and Technology*, 9:2395–0056, 04 2022.

[3] Dawn M. Sarno, Maggie W. Harris, and Jeffrey Black. Which phish is captured in the net? understanding phishing susceptibility and individual differences. 37(4):789–803. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/acp.4075.

[4] Giuseppe Desolda, Francesco Greco, and Luca Vigano. APOLLO: A GPT-based tool to detect phishing emails and generate explanations that warn users. 9(4):EICS003:1–EICS003:33.

[5] Nadjate Saidani, Kamel Adi, and Mohand Saïd Allili. A semantic-based classification approach for an enhanced spam detection. *Computers & Security*, 94, 2020-07.

[6] Sonowal and Gunikhan. Detecting phishing sms based on multiple correlation algorithms. *SN Computer Science*, 1, 11 2020.

[7] Panpan Zhang, Jing Ya, Tingwen Liu, Quangang Li, Jinqiao Shi, and Zhaojun Gu. imcircle: Automatic mining of indicators of compromise from the web. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019.

[8] Rasha Zieni, Luisa Massari, and Maria Carla Calzarossa. Phishing or not phishing? a survey on the detection of phishing websites. 11:18499–18519.