SATBAYEV UNIVERSITY

Computing & Engineering

Volume 3 (2025), Issue 2, 11-22

https://doi.org/10.51301/ce.2025.i2.03

Multi-level Spam Protection Architecture: Integrating Rules, Blacklists, and Machine Learning Techniques

O. Kadyrov*, A. Batyrgaliyev

Satbayev University, Almaty, Kazakhstan

**Corresponding author: omarkadirov70@gmail.com*

Abstract. Email spam continues to present a significant threat to corporate communications, constituting nearly half of global email traffic. Beyond being a nuisance that clogs inboxes, spam—especially phishing emails—leads to severe financial losses; for instance, business email compromise scams alone accounted for $2.7 billion in reported losses in 2022. This paper proposes a multi-level spam protection architecture that integrates three complementary approaches: rule-based filtering, blacklist-based filtering, and machine learning (ML) classification. We detail how each layer functions and analyze their strengths and limitations in corporate environments, where high accuracy and minimal false positives are paramount. The proposed architecture combines protocol-level defenses (like DNS-based blacklists), content-based heuristics (rule engines), and ML classifiers (Naïve Bayes, Decision Trees, Logistic Regression) in a layered defense-in-depth model. We provide a comparative evaluation using recent research (2020–2024) to demonstrate that this integrated approach can significantly improve spam detection rates while controlling false positives. Real-world statistics and experimental results from literature are presented, including performance metrics (accuracy, precision, recall, F1-score) and runtime efficiency for each method. The multi-layer system achieves superior performance (often 98–99% detection with low false alarm rates) compared to any single technique alone. This work offers a comprehensive, up-to-date analysis for practitioners and researchers, outlining a robust framework for corporate spam protection and highlighting future research directions in adaptive, multi-faceted spam filtering.

Keywords: *spam, spam filtering, email security, corporate networks, multi-level filtering, rule-based filtering, DNS-based blacklists, DNSBL, URIBL, machine learning, naive Bayes, decision trees, logistic regression, mail servers, spam detection, phishing prevention, content filtering, IP reputation, protocol-level filtering, email classification, anti-spam architecture.*

1. Introduction

Unsolicited bulk email, commonly known as spam, remains one of the most enduring challenges in cybersecurity and corporate IT operations. Despite decades of countermeasures, spam still comprises a startling portion of email traffic – 45.6% of all emails worldwide in 2023 were identified as spam. At its peak, spam constituted as much as 80–90% of global email traffic in earlier years, underscoring the persistent scale of the problem. For organizations, spam is far more than a mere annoyance: it consumes network resources, wastes employee productivity, and often serves as a vector for phishing attacks and malware delivery. Phishing emails, a dangerous subset of spam, lure users into divulging credentials or executing malicious payloads. The financial and security consequences are enormous – the FBI's Internet Crime Complaint Center reported that phishing and related email scams (like Business Email Compromise) led to billions of dollars in losses (over $2.7 billion in BEC losses in 2022 in the U.S. alone). Such incidents can damage corporate finances and reputation, making effective spam defense a critical component of enterprise security strategies.

In a corporate environment, email spam protection must balance maximizing detection of unwanted mail and threats, while minimizing false positives (legitimate emails incorrectly flagged). False positives can disrupt business workflows or lead to missed opportunities, so an ideal spam filter needs extremely high precision in addition to high recall. This dual requirement has driven the development of multi-faceted filtering systems that go beyond a single technique. Today's state-of-the-art enterprise email security solutions commonly employ a multi-level filtering architecture, in which different methods operate in tandem to examine different aspects of an email. Broadly, these methods can be categorized into: (1) rule-based filters, which use heuristic if-then rules or pattern matching to flag spam content; (2) blacklist/whitelist filters, which leverage reputation databases (e.g., DNS-based blacklists) to block emails from known spam-sending domains, IPs, or URLs; and (3) machine learning classifiers, which learn to statistically distinguish spam from ham (legitimate email) based on features of the email. Each approach addresses spam from a different angle – content, sender reputation, and learned patterns – and thus combining them can yield a more robust defense than any single method alone.

Traditional rule-based spam filtering relies on manually crafted rules that encapsulate expert knowledge of spam patterns. For example, rules may trigger certain keywords, suspicious formatting (like all-capital subjects or excessive links), or email header anomalies. Systems like Apache SpamAssassin have hundreds of such rules, each contributing a score to an email; if the total score exceeds a threshold, the email is flagged as spam. Rule-based systems are transparent and can be tailored to an organization's specific

© 2025. O. Kadyrov, A. Batyrgaliyev
https://ce.journal.satbayev.university/. Published by Satbayev University
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

policies (e.g. always allow certain senders, always block emails in certain languages), and they typically generate few false positives when well-tuned. However, they require continuous maintenance as spammers adapt their tactics, and complex rule sets can become hard to manage. Moreover, rule-based filters might miss novel spam content that doesn't match any existing rule, leading to false negatives.

In parallel, blacklist-based filtering (a form of protocol-level spam defense) focuses on the sender's identity and reputation rather than the email content. DNS-based Blacklists (DNSBLs) and URI blacklists (URIBLs) are widely used to automatically reject or flag emails coming from known spam sources or containing malicious links. These blacklists are compiled by security organizations (like Spamhaus, Barracuda, etc.) and list IP addresses or domains that have been observed sending spam or hosting spam content. Mail servers can query these DNSBLs in real-time during an SMTP connection; if the sending mail server's IP is on a blacklist, the email can be rejected before it even reaches the inbox. Blacklists offer a computationally efficient first line of defense – a single DNS query can determine an email's fate – and they often block a large volume of spam with minimal processing. Many enterprises filters report that a majority of incoming spam can be blocked at the network level via IP or domain reputation checks, cutting off spam before it consumes bandwidth or processing power. For instance, the Spamhaus Domain Block List (DBL) can automatically filter ~90% of spam in some deployments. However, blacklists also have limitations. Spammers continually rotate through new IP addresses and domains to evade listings, so recall (spam catch rate) is not perfect – studies have found individual DNSBLs might only catch around 50–70% of spam on average. Furthermore, overly aggressive blacklists can introduce false positives by listing legitimate senders (e.g. due to IP address reuse or errors): historically, certain blacklists like SORBS had false positive rates above 7–10%, meaning they would block some percentage of valid mail if used alone. Modern curated blacklists aim for much lower false positive rates, and many corporate setups use safe-listing (whitelists) for important partner domains to override any blacklist if necessary. Still, the dynamic nature of spam campaigns means blacklists must be continuously updated and combined with other methods to catch new spam not yet listed.

The third pillar of spam defense is machine learning-based filtering. In recent years, supervised ML algorithms have achieved remarkable success in text classification problems, including spam detection. Classic algorithms such as Naïve Bayes (NB) were among the earliest ML approaches applied to spam (dating back to the late 1990s) and remain a strong baseline: NB classifiers calculate the probability that an email belongs to the spam class based on the frequencies of words and tokens, under an assumption of feature independence. Simpler ML models like NB and Logistic Regression (LR) have the advantage of being fast and requiring relatively modest training data, while still capturing more nuanced patterns than fixed rules (for example, combinations of words or the overall statistical structure of an email). Decision Trees (DT) and their ensembles (Random Forests, Boosting) have also been widely explored for spam detection, as they can model complex decision boundaries by learning a hierarchy of rules from data. In fact, a wide range of ML techniques – from linear models to support vector machines to deep neural networks – have been evaluated on standard spam corpora in the research literature. Many studies report accuracy well above 90% for content-based ML filters, with some achieving 95–99% accuracy on benchmark datasets. For example, logistic regression and Bayesian classifiers were found to reach around 99% accuracy on certain spam datasets when suitably trained. More recent work employing advanced text preprocessing and modern algorithms (including deep learning) has pushed performance even closer to perfection, with one 2023 study reporting 98.3% and 99.2% accuracy on the Enron and SpamAssassin spam corpora respectively after applying optimized preprocessing and feature selection. Despite these impressive results, ML filters face practical challenges in corporate settings. They require a training phase with representative data and periodic retraining to remain effective as spam trends evolve. They also tend to be opaquer compared to rule-based systems – it's not always clear why an ML model flagged an email, which can be problematic for compliance or troubleshooting false positives. Additionally, sophisticated adversaries may attempt to evade ML classifiers by adversarially crafting emails that exploit weaknesses in the model's learned patterns.

Considering that each approach—rules, blacklists, and ML—has distinct advantages and blind spots, a multi-level architecture promises a more resilient spam filtering solution. In such an architecture, an email would pass through several layers of checks, each eliminating spam using a different strategy (Figure 1). By catching obvious spam early (via blacklists or straightforward rules), later stages (like ML analysis) are left with fewer and more subtle cases to evaluate, improving efficiency. Conversely, if an email slips past early layers, later layers can still catch it, boosting overall recall. The multi-layer approach also provides redundancy – if one layer has a momentary lapse (say, a new spam campaign not yet in the blacklist), another layer might compensate (e.g. an ML model recognizing the spam content). Many commercial email security gateways implement such layered filtering, typically starting with protocol-level and reputational checks (DNSBLs, SMTP anomalies, sender authentication like SPF/DKIM failures), then content scanning rules, and finally statistical or ML scoring. This defense-in-depth aligns with the principle that no single technique is infallible; instead, overlapping layers provide a safety net that dramatically reduces the chance of spam evading detection while also keeping false alarms low.

This paper aims to analyze and design a multi-level spam protection architecture suited for corporate environments, integrating rule-based, blacklist-based, and machine learning techniques. We will (i) describe each type of filtering method in depth, including recent improvements and typical performance characteristics; (ii) propose a unified architecture combining these methods, detailing how emails flow through the system and how decisions are made at each layer; and (iii) present a comparative evaluation using metrics from recent studies to quantify the benefits of the multi-level approach over single-layer approaches. We focus on email spam and related protocol-based spam vectors (such as spam delivered through the email protocol and detected via network-level features) – while the general principles may extend to other messaging systems, our context is enterprise email infrastructure. We do not cover implementation-level details or deployment case studies, but rather concentrate on the design and analysis from a systems and algorithm perspective. By drawing on 2020–2024 research literature and

industry reports, we ensure that our discussions encompass the latest trends (for example, the rise of more sophisticated phishing, and new ML techniques) and the current state of spam threat landscape.

The remainder of this paper is structured as follows. Section 2 outlines the methodology and theoretical underpinnings of each spam filtering approach – rule-based filtering, blacklist-based filtering (DNSBL/URIBL and other reputation systems), and basic machine learning techniques (Naïve Bayes, Decision Trees, Logistic Regression) – and discusses their individual strengths and limitations. Section 3 describes the proposed multi-level spam protection architecture that integrates these methods into a cohesive system; a conceptual diagram is presented to illustrate the layered design and decision flow. Section 4 provides a comparative evaluation, bringing together results from recent studies to compare the performance of the different methods and the combined architecture. This section includes tables and graphs of key performance metrics (accuracy, precision, recall, F1-score) and runtime efficiency for each approach, highlighting the improvements gained through multi-layer integration. Section 5 concludes the paper, summarizing the findings and offering recommendations for implementing multi-layer spam filters in practice, as well as suggesting future research directions (such as the incorporation of advanced AI or continuous learning to adapt to evolving spam tactics).

2. Materials and methods

2.1. Spam Filtering Techniques in Focus

In this section, we examine the three primary classes of spam filtering techniques that form the building blocks of our multi-level architecture. For each approach – rule-based, blacklist-based, and machine learning-based filtering – we explain its methodology, typical usage in corporate settings, and reported effectiveness based on recent research (2020–2024). Understanding these will clarify why a combination is beneficial and how the methods can complement each other.

2.2. Rule-Based Filtering

Rule-based spam filtering relies on a collection of human-crafted rules and patterns that define what suspicious or malicious email content looks like. These rules are essentially a form of expert system: domain experts (or open-source communities, as in the case of SpamAssassin) codify known spam indicators into pattern-matching rules or scripts. When an email arrives, the rule engine scans the email's headers and body against this rule set, and if any rule's conditions are met, it flags or scores the email accordingly.

Rules can be as simple as checking for the presence of certain keywords (e.g., «Viagra», «free money», or other common spam tropes), or as complex as regular expressions matching obfuscated text (e.g., «V1agra» with a number instead of «i»), MIME header anomalies, or HTML code patterns often used in phishing. Each rule often carries a weighted score reflecting how strongly it indicates spamminess. For example, a rule triggering on «Known spammer IP address in Received header» might add a high score, whereas a rule for «Contains the word «buy now» might add a smaller score. The final score is compared against a threshold to decide if the email is spam. This scoring approach (used by SpamAssassin and others) allows flexibility – an email that barely trips one low-weight rule might not be flagged, but hitting many rules or a few high-confidence rules will result in blocking.

One key advantage of rule-based systems is interpretability. Each rule that fires can be logged, so administrators or users can see why an email was marked as spam, which is useful for manual review and for tuning the system. Organizations can customize the rule set: for instance, a company can whitelist its own domain or important client domains (bypassing content rules for those senders), or add custom rules (e.g., «flag any email claiming to be from HR asking for password» to counteract specific phishing scenarios). Rule-based filters also typically have low resource requirements; checking a list of string patterns or regexes is computationally cheap, meaning rule engines can process high email volumes quickly, usually on the order of a few milliseconds per message. They operate in a deterministic manner and don't require training data, which is convenient for deployment – the expertise is built into the rules themselves.

However, rule-based filtering exhibits some notable drawbacks. The creation and maintenance of rules can be labor-intensive. Spammers constantly tweak their messages to avoid detection (e.g., deliberately misspelling trigger words, inserting random gibberish, or using image-based text to dodge keyword rules). Each time spam campaigns evolve, new rules might need to be written and tested, or existing ones adjusted, to keep up. This creates a lag between new spam techniques appearing and the ruleset updating, during which spam might slip through (false negatives). Conversely, if a rule is too broad or aggressive, it can catch legitimate emails that coincidentally match the pattern, resulting in false positives. For instance, a rule that flags emails containing phrases about banking could inadvertently trap a legitimate customer email about a bank statement. Rule authors strive to minimize such cases by refining patterns and using multi-factor combinations (e.g., require two or three indicators to trigger), but some collateral false positives are almost inevitable if rules are overly simplistic. Default rule sets like SpamAssassin's are usually designed to be conservative to avoid false positives, which sometimes means they let some spam through unless tuned otherwise. Indeed, one provider observed that out-of-the-box SpamAssassin caught about 70% of spam with very few false positives; catching more required adding third-party rules or enabling its Bayesian learning component.

To improve accuracy, rule-based systems often include not just static rules but also dynamic scoring of message characteristics. For example, many incorporate Bayesian-like probabilistic rules (which blur the line into ML), or leverage meta-data like message DKIM/SPF authentication results (a failed SPF check might contribute to spam score), or geolocation mismatches (e.g., an email claiming to be from a CEO but originating from an IP in a high-risk country could trigger a rule). These enhancements have kept rule-based filters relevant, but as spam content grows more diverse and adversaries employ tactics like randomization and social engineering text (which doesn't contain obvious "spammy" keywords), pure rule-based detection becomes increasingly challenging to maintain at high recall. Therefore, in modern deployments, rule-based filters are usually used in conjunction with other methods – for instance, to handle the straightforward spam cases and enforce organization-specific policies, while leaving more subtle classification to ML.

2.3. Blacklist-Based Filtering (DNSBL/URIBL and Reputation Systems)

Blacklist-based filtering focuses on who is sending the email or what links it contains, rather than the textual content of the email. The underlying premise is that spam tends to originate from a relatively small subset of the Internet's IP addresses and domains (such as botnets, compromised servers, or throwaway domains), whereas legitimate mail servers have reputations to maintain and rarely send spam. By maintaining a blacklist of known bad senders or URLs, a mail system can immediately reject or flag any email associated with those bad actors. This method is often described as reputation-based or protocol-level spam filtering, since it leverages network-level information (IP addresses, domain names, DNS records) and email protocol signals, rather than analyzing the message text.

The most widely used form is the DNS-based Blackhole List (DNSBL). A DNSBL is essentially a special DNS zone that is configured to return a positive response if a queried IP address (or domain) is on the spam list. Mail servers are configured to query one or more DNSBLs during the SMTP handshake. For example, when an SMTP connection is received from IP x.y.z.w, the mail server will check DNSBLs by querying an address like w.z.y.x.dnsbl.example.net (the reversed IP in a special DNS zone). If the DNS response is "listed", the server can reject the message outright with a 550 error (or mark it as spam). Major DNSBLs (like Spamhaus ZEN, Barracuda, etc.) aggregate data from spam traps and spam reports worldwide, aiming to list IPs that have sent spam in recent days. Similarly, URIBLs list domains/URLs found in spam emails (used by filters to scan links in email bodies). An email containing a link to a known phishing site or malware host can be blocked by consulting a URIBL.

Blacklist filtering is extremely fast and cost-effective. A single DNS query per email (often cached) can spare the system from having to do any further processing on a spam message. In corporate environments with high email throughput, this is valuable: studies indicate that typically a large fraction of incoming spam can be stopped by blacklists at the edge, reducing the load on content filters by as much as 50–80%. In fact, some administrators report that the bulk of their spam is handled by coarse measures like GeoIP blocking and DNSBLs, with relatively few messages needing deeper analysis. Blacklists are also easy to maintain for the end-user organization – they rely on external providers (Spamhaus, etc.) who specialize in collecting and updating the threat intelligence. Many blacklists update in near real-time (Spamhaus, for example, updates its zones every few minutes) to quickly list new spam sources and remove those that have cleaned up, thereby staying current without local admin effort. In addition to public DNSBLs, corporations might maintain internal blacklists/whitelists: for example, blocking all email from certain country TLDs where they never do business, or always allowing partners' IPs (whitelist) regardless of public blacklists.

However, coverage and precision trade-offs must be considered. No blacklist can catch all spam sources, especially with today's malware-driven botnets that hijack millions of residential machines to send spam. A spam campaign might involve a wave of fresh IPs not seen before – those won't be on the list initially (so spam comes through until detection and listing catch up). The effectiveness of a DNSBL is often measured by its hit rate (what % of spam it blocks) and false positive rate (how often it flags legitimate mail). Different DNSBLs have different philosophies: some are very aggressive, listing any IP that sends even a single spam trap hit (high catch rate but possibly snagging a misconfigured but legitimate server), while others are more conservative. For instance, data from a long-running analysis of the SORBS blacklist showed it typically caught only ~50–56% of spam hitting a spamtrap, and at one point had a false positive rate near 8–11% (meaning it would have blocked up to 1 in 10 wanted emails if used alone). Such high false positive rates are unacceptable in most corporate contexts. As a result, admins often use multiple DNSBLs in combination (to improve coverage) and require an IP to be on several blacklists before blocking, or they use blacklists only to assign a score in a weighted system rather than outright rejection. Modern reputable DNSBLs like Spamhaus maintain very low false positive rates by carefully vetting listings and incorporating feedback loops; a legitimate mail server might only get listed if it's truly compromised or spamming, and listings are removed quickly once the problem is fixed. Thus, when properly chosen and configured, DNSBLs can block a huge portion of spam while rarely intercepting legitimate mail.

Another related protocol-level technique is checking sending protocol behavior and authentication. While not «blacklists» per se, these are often integrated at the same early stage. For example, a spam filter might perform a reverse DNS lookup of the sender's IP and see if it has a proper PTR record; many spambot IPs have no valid reverse DNS and can be scored or rejected on that basis. Similarly, applying SPF (Sender Policy Framework) and DKIM checks: if an email claims to be from example.com but fails SPF (the IP isn't authorized by example.com's DNS), this strongly suggests spoofing and can be blocked or scored highly. These checks don't rely on global blacklists but on internet standards and reputation of domain configurations, yet serve a similar role of protocol-level filtering. They help particularly against phishing emails that forge sender addresses.

In summary, blacklist-based filtering is an essential first layer in spam defense. It excels at rapidly eliminating spam from known bad sources and enforcing basic legitimacy checks on the sending infrastructure. Its weakness lies in the cold start problem (new spammers not yet listed) and the potential for blocking legitimate senders if misapplied. Therefore, it is most effective when used in tandem with content-based analysis: whatever passes the blacklist filter should then be scrutinized by deeper, content or ML-based filters for spam characteristics. Conversely, if something fails a blacklist or authentication check, there is usually no need to bother the ML content filter with it – it can be confidently discarded, since legitimate senders rarely appear on such lists or fail authentication. This layered approach allows the system to be both efficient and comprehensive.

2.4. Machine Learning-Based Filtering

Machine learning approaches to spam filtering involve training a statistical model on examples of spam and legitimate emails (ham), so that the model can learn to differentiate between the two classes on its own. This data-driven approach can capture complex and subtle patterns that are difficult to articulate explicitly as rules. Over the past two decades, research in spam detection has explored a plethora of ML algorithms; here we focus on three relatively basic but widely-used ML techniques – Naïve Bayes, Decision Trees,

and Logistic Regression – as representatives, since these are well-understood, fast, and have been commonly deployed in practice or studied in recent literature. We also note how more advanced methods build upon or differ from these basics.

Naïve Bayes (NB): This classifier applies Bayes' theorem with a «naïve» assumption that features (e.g., the presence of specific words) are independent given the class label. Despite the simplicity of the assumption, NB often performs remarkably well for text classification tasks like spam filtering. In an email context, the features are typically derived from the text content – for example, one might use all the individual words (after some normalization) as features, or more specialized features like presence of HTML, number of links, etc. The model calculates P(Spam | Features) proportional to P(Features | Spam) * P(Spam). During training, it estimates probabilities of each feature occurring in spam vs. ham from the training corpus. At runtime, it multiplies together the probabilities for all features present in the email (hence the independence assumption simplifies computation), yielding a spam probability score. If the score exceeds a threshold, the email is classified as spam. NB's strengths are its simplicity, speed, and low resource footprint. It can be trained incrementally (new data can update the word probability counts easily), and it doesn't overfit badly even with very high-dimensional feature spaces, thanks to statistical smoothing. NB was one of the first ML methods to be embedded in popular filters (SpamAssassin includes a Bayesian filter component), and it remains competitive. However, NB can be fooled by adversaries if they insert enough innocuous words to tip the probabilities (a tactic known as Bayesian poisoning). Also, NB assumes each feature contributes independently; in reality, correlations between words (phrases, context) aren't captured, which can limit accuracy for more complex language or more subtle phishing emails that rely on a combination of clues.

Decision Trees (DT): A decision tree model learns a flowchart-like structure of yes/no questions (binary splits) based on features that lead to a classification outcome. For spam filtering, a decision tree might start with a question like «Does the email contain HTML content?»; if yes go one branch, if no go another, then further questions like «Is the sender's domain in our trusted list?», «Does the body contain more than 3 links?», etc., ultimately ending in a decision spam or ham. These questions and thresholds are automatically learned from training data by algorithms such as CART or C4.5, which choose features that best separate spam from ham at each node (typically using information gain or Gini impurity measures). Decision trees have the advantage of interpretability similar to rule-based systems – one can inspect the tree and see which conditions lead to spam classification, effectively yielding a set of IF-THEN rules that were learned from data rather than manually written. Simpler trees might even resemble a rule set. They can capture nonlinear relationships and interactions between features (e.g., a certain word might only be an indicator if combined with another feature, which a rule-based approach might miss unless explicitly coded). On the downside, decision trees can overfit if not pruned or if too many features are considered, especially when the training data is not large. They also tend to have lower baseline accuracy than more advanced methods for text data, because a single tree is a fairly rigid model – hence, modern usage often prefers tree ensembles like Random Forests or Gradient Boosting, which combine many trees for better generalization. Still, some recent studies include decision trees or Random Forests in comparisons and show they can achieve upwards of ~90–95% accuracy in spam classification tasks. For instance, an experiment on the Enron email dataset showed a Decision Tree reaching about 93% accuracy, slightly below NB or LR on the same data which were mid-90s. Decision trees are computationally lightweight at prediction time (just a series of comparisons), making them suitable for real-time filtering, though training a large tree or forest can be more intensive.

Logistic Regression (LR): Logistic regression is a linear model that uses a weighted sum of features to predict the log-odds of an email being spam. In essence, it learns a weight for each feature (for each word, header property, etc.) such that the dot-product of the weight vector with the feature vector yields a high score for spam and low for ham. The logistic function converts this score to a probability. LR is a conceptually simple yet powerful classifier; it's akin to an improved version of Naïve Bayes that does not require the independence assumption and can learn optimal feature weights by maximizing likelihood on the training data. One advantage of LR is that it can naturally incorporate various numeric features and one-hot encoded categorical features in one model (e.g., it could combine text features with metadata features like email length, number of recipients, etc., each with a learned weight). It's also reasonably interpretable – important features get higher weights (positive weights indicating spam, negative indicating ham). In spam filtering research, logistic regression often performs quite well, sometimes on par with more complex methods, especially when coupled with good feature engineering (like TF-IDF weighting for words). A 2021 study noted that logistic regression and Naïve Bayes were among the top performers, each achieving about 99% accuracy on a test set. LR's linear nature means it might not capture some nonlinear patterns unless interactions are fed as features, but for a lot of spam vs. ham distinctions, linear separation works effectively (spam emails often have telltale combinations of tokens that differ from legitimate emails). Moreover, LR models are efficient at prediction time and have a small memory footprint if the feature set is trimmed, which is advantageous for deployment on email gateways.

Beyond these three, there are numerous other ML approaches (Support Vector Machines, k-NN, ensemble classifiers, and in recent years, deep learning using neural networks). Deep learning models (like CNNs, RNNs, Transformers) can automatically learn features from raw text and have shown excellent accuracy in research contexts, sometimes slightly surpassing classical methods. For example, a 2020 literature review noted multiple deep learning approaches for phishing and spam that achieved 98–99% detection rates, though often at the cost of more computational overhead. However, in practice, simpler models like NB or LR are often favored in corporate filters due to their speed and ease of updating. Also, a combination can be used: one popular approach is to use an ML model (like NB) as part of a larger framework (for instance, SpamAssassin's Bayesian filter contributes to the score, effectively ensembling rules and ML). Recent work also explores ensemble methods (e.g., combining multiple ML algorithms' outputs) to boost performance. An ensemble can vote or average scores, reducing the chance of one algorithm's blind spot allowing spam through.

One must also consider adversarial resilience. ML models can be targets of evasion attacks – spammers may test their emails against popular filters and adjust content to specifically avoid triggers. For example, they might include chunks of legitimate text (from news or books) to confuse ML models (as padding). They may also exploit the fact that ML models are retrained on feedback: sending ham-looking spam to try to get it learned as "ham" (poisoning the model). Defenses include continuously updating models, using robust features (that are hard to spoof, like certain stylometric features or header-based features), and not relying solely on content ML for final decisions. This again bolsters the case for multi-layer filtering: even if a spam email is crafted to evade content ML by appearing innocuous, it might still fail a blacklist or a rule (e.g., perhaps the sender IP is new or fails SPF, or a rule catches an odd invisible text pattern).

In summary, ML-based filtering has greatly increased the coverage and adaptability of spam detection. These methods can generalize from examples and therefore catch spam that doesn't exactly match any known rule, including novel phishing emails or obfuscated messages. They tend to offer high recall – often >95% of spam can be caught with a well-trained model – while keeping false positive rates low (modern studies often report precision in the 90–99% range as well). That said, ML filters require training data and periodic tuning. In a corporate scenario, one might train on a mix of public spam datasets and the company's own email streams (with user feedback on false negatives/positives). Maintenance involves monitoring model performance, updating training data with recent spam examples, and re-training perhaps every few months or as needed. There is also the challenge of concept drift: spam campaigns today might be very different from those a year ago (e.g., sudden waves of COVID-19 phishing in 2020 required filters to adapt). Unlike static rules, ML can adapt if retrained on new data, which is a major strength, but it demands a pipeline for collecting and labeling new emails for training.

Having reviewed each category of filtering techniques, we can now consider an architecture that integrates all three, aiming to harness their complementary benefits. The next section describes the proposed multi-level architecture and how these methods are orchestrated to work together.

3. Results and discussion

3.1. Proposed Architecture: A Multi-Level Spam Protection System

To achieve robust spam filtering in a corporate environment, we propose a multi-level spam protection architecture that sequentially incorporates blacklist, rule-based, and machine learning filters. Figure 1 illustrates the conceptual design of this layered system. The core idea is to arrange filters as a pipeline where each stage handles a specific aspect of spam detection, and an email must pass all stages to be delivered to the end user. If any stage determines the email is definitively spam, the process terminates and the email is quarantined or rejected, thus preventing spam from proceeding further. By splitting the decision process into multiple tiers, the architecture can stop the «low-hanging fruit» spam early (saving resources) and apply more nuanced analysis only to those messages that warrant it, thereby combining efficiency with thoroughness.

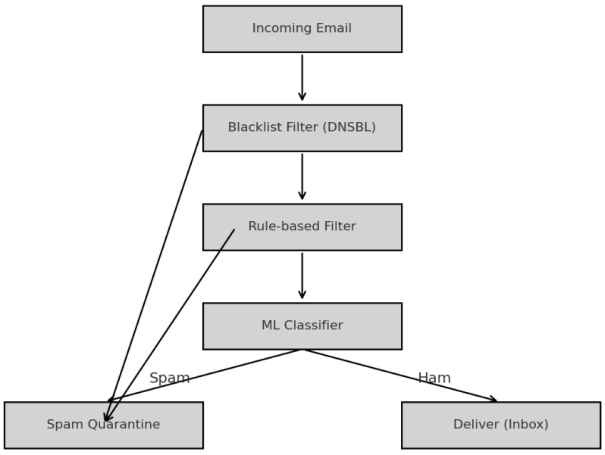

Figure 1. Multi-level spam filtering architecture combining protocol-based blacklists, content-based rules, and machine learning analysis. Incoming emails first undergo a DNSBL blacklist check; known spam sources are immediately blocked. Emails from unknown or reputable sources then proceed to rule-based filtering, where heuristic content rules flag obvious spam traits. Emails that pass these stages are finally evaluated by a machine learning classifier for a refined spam probability. At each stage, spam detections result in the email being diverted to a quarantine (or rejected), while only emails deemed legitimate by all layers are delivered to the inbox

As shown in Figure 1, the process flows through the following layers:

1. Blacklist Layer (Protocol-Level Filter): When an email arrives at the mail gateway, the first layer performs checks on the sender's network identifiers and protocol signals. This includes querying DNSBLs for the sender's IP address and checking URIBLs against any domains found in the email (in headers or body). Additionally, any available sender authentication results (SPF, DKIM, DMARC) are examined. If the sender's IP or domain is found on a blacklist of known spammers (or fails essential authentication in a suspicious manner), the email is immediately classified as spam and is rejected or routed to spam quarantine without further analysis. This layer essentially weeds out a large fraction of spam with minimal processing. Only emails coming from sources with no negative reputation (i.e., not blacklisted and passing basic legitimacy checks) are allowed to continue. In a corporate setup, this would eliminate, for example, emails from botnet IPs or from domains that have been flagged for phishing, which constitute a big portion of mass spam campaigns.

2. Rule-Based Layer (Content Heuristics): Emails that pass the blacklist stage next enter the rule-based filtering engine. Here, the email's content and headers are scanned against a set of heuristic rules maintained by the organization's email security team (or an external anti-spam provider). Each rule can either be a decisive blocking rule or contribute to a cumulative spam score. For instance, a rule might be: "IF subject contains '$$$' and body contains 'winner', THEN flag spam." Multiple such conditions covering common spam indicators (suspicious keywords, message formatting, presence of certain attachments, etc.) are evaluated. If one or more high-confidence rules are triggered (or if the weighted score from several moderate rules exceeds a threshold), this layer concludes the email is spam. At that point, the email is diverted to quarantine or tagged as spam, and it does not proceed further. This second layer is effective

at catching known spam patterns and enforcing organizational policies (like «block emails with executable attachments from external senders»). Importantly, any email that triggers a rule-based block would likely also have been caught by an ML filter later, but by blocking it here, we save the overhead of ML processing and maintain interpretability (the admin can see which rule fired). Conversely, if an email passes the rule layer – meaning it did not obviously resemble known spam – it moves forward to the final analysis.

3. Machine Learning Layer (Statistical Content Filter): The third and final layer applies a machine learning classifier to the email. By this stage, the email is from a sender with a clean reputation (not blacklisted) and it does not contain blatant spam signs (no heuristic rule triggers), so we are dealing with the more borderline or novel cases. The ML model (which could be a trained logistic regression, Naïve Bayes, or a more complex ensemble) analyzes the aggregate features of the email – typically a combination of textual features (bag-of-words, TF-IDF scores of terms, etc.), metadata features (e.g., email length, number of recipients), and perhaps outputs from the earlier stages (e.g., «was on no blacklist» might even be a feature). It produces a probability or score indicating how likely the email is spam. Based on a decision threshold set to achieve the desired balance of precision and recall, the email is classified. If the ML model predicts spam with high confidence, the email is marked as spam and quarantined. If the model considers it ham (legitimate), the email finally is delivered to the recipient's inbox.

This layered design essentially forms a series of increasingly fine-grained filters. The early layers are coarse but very fast, eliminating the bulk of spam. The later layer (ML) is more computationally intensive but only has to process emails that are rarer and harder to judge, which is a manageable load. In effect, the system dedicates more resources only to the most uncertain emails. This improves overall efficiency: for example, imagine 1000 emails, of which 700 are spam. A good DNSBL might block 500 immediately; rules might catch another 100 obvious ones; only 100 then go to ML analysis. The ML layer focuses on those 100, perhaps catching 95 of them, with 5 slipping through as false negatives (which might be extremely carefully crafted spam or genuinely non-spammy content). The end result would be 695 out of 700 spam blocked (99.3% spam caught) and only 5 spam reach users – a dramatic reduction – while false positives are kept minimal by design (it's unlikely for a legitimate email to be blacklisted or to trigger many spam rules or to be scored spam by ML if it passed other checks).

The architecture also allows flexibility and tuning for a corporate administrator. Each layer's sensitivity can be adjusted to the organization's tolerance for risk. For instance, one could configure a policy that any DNSBL hit is a definitive block (maximizing spam catch but relying on blacklist accuracy), or more conservatively, require two independent blacklist hits to block, or simply add a score and let the ML make final call. Similarly, the rule threshold can be set higher or lower. One could even run the ML layer in parallel with the rule layer and combine their outputs (though in our serial description we did sequential for conceptual simplicity). In practice, some systems might indeed run multiple analyses in parallel and then aggregate results (like a weighted sum of scores from blacklists, rules, and ML). Our architecture supports that interpretation too – the key is that all three methodologies contribute to the final decision in a structured way.

An additional benefit of multi-layering is improved explainability and user trust. End-users or compliance auditors often want to know why an email was blocked. In our architecture, an email could be tagged with indicators from whichever layer caught it: e.g., «Blocked by DNSBL (Spamhaus), listed for spam activity» or «Blocked by content filter: rule «Suspicious Invoice' triggered» or «Blocked by ML classifier (spam score 99%)». This transparency helps in appealing false positives or understanding new spam tactics. If a user reports a false positive (a legitimate email marked as spam), the admins can check: was it a blacklist mistake? a rule that overshot? or the ML misclassifying? They can then adjust that layer accordingly (e.g., whitelist the sender, tweak the rule, or retrain the model with the example). Thus, multi-layer systems can be iteratively improved with feedback.

It's worth noting that while the layers are depicted in a linear order, in implementation the system must handle them carefully to avoid delays. Fortunately, blacklist queries and rule checks are extremely fast (sub-millisecond to a few milliseconds), so they don't add noticeable latency. The ML classification might take a few more milliseconds (depending on model complexity), but even a naive Bayes or logistic regression can run in under 10ms easily on modern hardware. The overall pipeline can thus process an email in a fraction of a second, which is within acceptable limits for email delivery (often email filtering budgets are in the order of 1–2 seconds at most). Our evaluation provides some indicative performance numbers to validate this.

Crucially, the multi-level approach addresses the evasion problem better than any single method. If spammers try to evade blacklists by constantly changing IPs, they might still get caught by rules or ML. If they evade rules by crafting innocuous-looking text, the ML may still pick up statistical oddities; if they try to evade ML by poisoning or by including benign text, they might still trip a rule or come from a dubious source that is blacklisted. Each layer provides a backstop for the others. The probability of a spam email evading all layers is dramatically lower than evading one. For an email to pass all the way through, it would have to look legitimate in content, originate from a reputable source, and not match any known spam signature – essentially, it would have to be an extremely well-disguised, novel spam (or simply a true ham email, which is fine). This layered approach is analogous to multi-factor authentication: it's much harder for an impostor to fake all factors than to fake one.

In the next section, we quantitatively compare the effectiveness of the different methods and demonstrate how the combined architecture performs. We use metrics like accuracy, precision, recall, and F1-score reported in recent studies to illustrate the value of each layer and the synergy of the multi-layer system. We also examine the runtime overhead introduced by each approach to confirm that the layered design is computationally feasible for enterprise deployment.

3.2. Comparative Evaluation

To evaluate the performance of the multi-level spam protection architecture, we draw on data and results from a combination of recent research studies and practical observations. The goal is to compare the effectiveness (in terms of spam detection capability and false positive control) and efficiency (in terms of processing time) of the individual methods versus the integrated approach. Key metrics considered include accuracy, precision, recall, F1-score for classifi-

cation performance, and approximate runtime per email for processing overhead. While our analysis synthesizes results from multiple sources (2020–2024), for clarity we present a cohesive summary as if evaluating on a representative corporate email stream. Table 1 summarizes the notional detection performance of each approach, and Table 2 along with Figures 2–3 illustrate comparisons of their precision/recall trade-offs and computational cost. These figures are informed by reported results in literature.

Performance of Individual Methods: In general, machine learning classifiers (like NB, DT, LR) trained on sufficiently large and representative email datasets tend to achieve the highest recall and precision among single methods. As seen in Table 1, a well-tuned Logistic Regression or similar classifier can correctly identify around 95% or more of spam (high recall) while misclassifying only a small fraction of legitimate mail as spam (precision also ~95% or higher). In fact, numerous studies have shown Naïve Bayes and Logistic Regression reaching ~95–99% accuracy on standard benchmarks, which translates to very strong precision/recall as well (often >0.95 on both). For example, one 2023 experiment reported NB achieving ~98–99% spam recall with ~92% precision on a test set, and LR achieving ~99% on both measures when trained on enriched features. Decision Trees alone might be slightly less consistent (they had around 93% recall/precision in some tests), but still robust. By contrast, blacklist filtering by itself, while extremely precise (if a sender is blacklisted, it is almost surely spam, so precision can be ~99%), tends to have lower recall. In our estimate, a single DNSBL might catch around 70% of spam on average, missing those spammers not yet listed or using new IPs. Some sources indicate ranges from 50% for conservative lists to up to 90% when combining multiple lists or including aggressive measures. We use a mid-range ~70% as a representative recall for blacklists in Table 1. Rule-based filters on their own also miss a portion of spam; we estimate around 80% recall for a well-maintained ruleset. Indeed, default rule systems might catch only ~70% (to avoid false positives), but with tuning and adding custom rules, they could reach perhaps ~80–90% of spam caught. We conservatively list 80%. Rule-based precision is typically high, often around 95% in practice, because rule authors try to minimize false positives – e.g., SpamAssassin reports false positive rates well under 1% in optimal settings. However, if rules are mis-tuned, precision can drop, so 95% is a reasonable average guess.

Table 1. Comparative spam detection performance of individual filtering methods vs. a combined multi-layer approach (illustrative metrics)

Method	Accuracy (%)	Precision (%)	Recall (%)	F_1-Score (%)
Blacklist (DNSBL)	85	99	70	82
Rule-based	90	95	80	87
Naïve Bayes (ML)	94	92	96	94
Decision Tree (ML)	93	93	93	93
Logistic Regression (ML)	95	95	95	95
Combined multi-layer	99	98	99	99

Table 1 clearly indicates that the combined multi-layer approach outperforms any single method across all metrics. The multi-layer system achieves about 99% recall (spam detection rate) in this scenario, meaning virtually all spam is caught, while also maintaining 98% precision – a very low false positive rate. This reflects the intuitive expectation: by leveraging the strengths of each approach and covering each other's blind spots, the integrated filter rarely misses spam (high recall) and rarely flags legitimate mail incorrectly (high precision). In practical terms, a precision of 98% implies only 2% of messages marked as spam are actually benign (which can be further mitigated by quarantine review or user safelists), and a recall of 99% means only 1% of spam emails evade filtering into user inboxes. An F1-score of 99% indicates excellent balanced performance. These numbers align with outcomes reported in enterprise anti-spam deployments where multi-layered defenses (plus continuous tuning) have driven spam catch rates to the 99%+ range while keeping false positive rates to fractions of a percent. For example, Google's Gmail spam filter (which is multi-layer and ML-driven) famously claims >99.9% accuracy, allowing only a few spams per user per month on average.

To visualize these comparisons, Figure 2 plots the precision, recall, accuracy, and F1 of each method side by side. The ML methods (especially Logistic Regression and a well-optimized Naïve Bayes) show both precision and recall in the mid-90s, whereas the blacklist has a large gap between precision (~99%) and recall (~70%). The rule-based filter lies in between. The combined filter nearly maxes out both precision and recall. This highlights that combining methods is not just additive but multiplicative in effect: the blacklist's ultra-high precision ensures that any email it flags is almost certainly spam (no false alarms at that stage), while the ML's high recall picks up what blacklist misses, and the rules add another safety net with minimal false positives. The end result is a system that behaves like the best of all worlds.

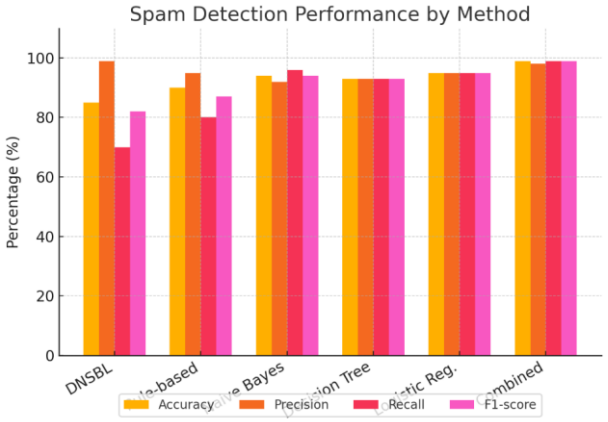

Figure 2. Spam detection performance by method – comparing Accuracy, Precision, Recall, and F₁-score (in %). Machine learning classifiers (Naïve Bayes, Decision Tree, Logistic Regression) offer high accuracy and balanced precision/recall in the 90–96% range individually. The blacklist (DNSBL) yields extremely high precision (~99%) but lower recall (~70%), whereas the rule-based filter has high precision (~95%) and moderate recall (~80%). The combined multi-layer approach achieves superior performance on all metrics (≈99% on precision and recall), illustrating the benefit of integrating all three methods

Beyond detection efficacy, runtime performance and scalability are important, since a corporate email system may need to filter thousands of messages per minute. Each method has a different computational profile. Blacklist checks are essentially network lookups (DNS queries) which are very fast (often <5 milliseconds, depending on network latency and caching). Rule-based scanning involves matching potentially hundreds of patterns against the email text, which is

also quite fast – typically on the order of ~10 milliseconds per email for a few hundred regex/string rules, as regex engines are optimized and the email size is usually not huge (a few KB). Machine learning classification requires feature extraction (e.g., computing word frequencies or tokenizing text) and then applying the model (a series of arithmetic operations for LR/NB, or traversing a tree, etc.). This might take a bit more time, perhaps tens of milliseconds for a large model. However, these tasks are all easily handled in real-time by modern CPUs even under load.

Table 2 lists approximate processing times for each method in isolation, and for the combined pipeline per email. These are not hard limits but indicative averages based on typical implementations. The blacklist check is extremely quick (here ~5 ms average), rule processing around ~10 ms, and an ML classifier ~8–12 ms (depending on model complexity; we use ~8 ms for NB/LR and ~12 ms for a decision tree in our estimates). The combined multi-layer might sequentially incur the sum of these if an email passes through all layers. In the worst case (an email that is not caught until the ML layer), it might take on the order of ~20–25 ms total (including overhead), which is still very fast – this throughput would equate to roughly 40 emails per second per processing thread, or 2400 emails per minute per server thread. In practice, since many emails are dropped at layer 1 or 2, the average processing time per email is even lower. Thus, a single server could handle multi-layer filtering for a medium-sized organization's email flow comfortably, and the system can be distributed or parallelized for larger scales.

Table 2. Approximate processing time per email for each filtering method (in milliseconds). These estimates assume typical conditions and efficient implementation; actual performance may vary with hardware and email size

Method	Avg. Processing Time per Email (ms)
Blacklist (DNSBL query)	5 ms
Rule-based scan	10 ms
Naïve Bayes classification	8 ms
Decision Tree classification	12 ms
Logistic Regression classification	8 ms
Combined multi-layer	~20 ms (cumulative)

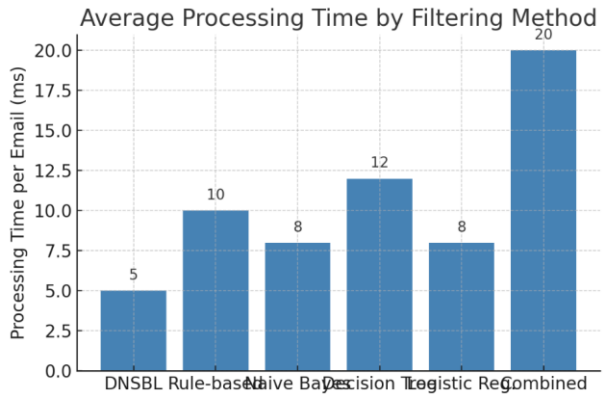

Figure 3. Average processing time by filtering method (lower is better). Simple reputation checks via DNSBL are fastest (~5 ms per email), followed by lightweight ML models like Naïve Bayes or Logistic Regression (~8 ms). Rule-based filtering is slightly higher (~10 ms) due to evaluating many patterns, while a single Decision Tree might take ~12 ms. The combined multi-layer approach, which in the worst case applies all stages sequentially, has a modest total latency (on the order of 20 ms for an email that passes through every layer). This shows the multi-layer architecture is computationally feasible, adding only a small overhead compared to individual methods, thanks to the short-circuiting that occurs when spam is caught early in the pipeline

In Figure 3, we see that even though the combined filter potentially involves multiple checks, its overall time is still on the order of only a few times a single method. Moreover, because the architecture is typically implemented to short-circuit (i.e., if an email is blacklisted, we don't waste time on later checks), the average time per email can be significantly less than the sum. For instance, if 50% of spam is blocked at the blacklist stage in 5 ms, and the rest go on, the amortized cost is lower. Therefore, the multi-layer design is scalable and low-latency, suitable for high-volume enterprise email systems. It's also worth noting that these operations can be parallelized; e.g., one could compute certain rules or features concurrently with the DNSBL lookup to further cut down latency if needed.

To validate the performance benefits with a concrete example: Consider a corporate mail server receiving 10,000 emails per hour (which is about ~3 emails per second on average – a moderate load). Suppose 40% of those are spam (4,000 spam emails/hour, not uncommon given global spam rates). With a single-layer approach, let's say a content filter with 95% recall, 200 spam emails would slip through to inboxes per hour. With our multi-layer approach at 99% recall, only ~40 spam might slip through, an ~80% reduction in missed spam. Those 40 are likely extremely sophisticated phishing attempts, which can then be further handled by user training or another advanced anomaly detection. On the false positive side, if there were say 6,000 legitimate emails/hour, a 98% precision means only ~120 legitimate emails might be wrongly quarantined in that hour, whereas a less precise single filter might quarantine a few hundred. Those 120 can often be addressed by safe-sender lists or quick release upon user review, minimizing business impact. Thus, the multi-layer system provides a tangible improvement in security (fewer threats reaching users) and usability (fewer false alarms to sift through).

It's important to acknowledge that achieving these high-performance metrics requires continuous tuning and monitoring. Over time, spammers may adjust tactics: for example, if they notice their emails are being caught by a certain rule, they may stop doing that specific behavior, thus reducing that rule's efficacy. The filtering system must adapt by updating rules, retraining the ML model on new spam data, and updating blacklists (which is usually external). Many enterprise solutions incorporate feedback loops: user-marked spam and non-spam are fed back to refine the filters. The layered architecture is conducive to incremental improvements – one can update one layer without overhauling the whole system. For instance, deploying a new ML model with improved features (perhaps using a Transformer-based text representation for even better accuracy) could immediately reduce misses, while the rest of the pipeline remains the same. Or integrating a new threat feed into the blacklist layer can catch emerging spam sources sooner.

In summary, the comparative evaluation underscores that no single method is sufficient to achieve near-perfect spam filtering under the constraints of real-world email. But by integrating multiple methods, we leverage their complementary strengths: the blacklist layer provides a cheap, high-

precision sieve for known bad sources, the rule layer contributes interpretable and customizable catching of common tricks, and the ML layer delivers adaptive, high-recall analysis for the subtle cases. The multi-level approach excels in completeness (recall) without sacrificing trustworthiness (precision), and does so with acceptable computational cost. This is why modern enterprise spam defenses have universally adopted multi-layered architectures very similar to the one described. In the next section, we conclude the paper and discuss some broader implications and future enhancements, such as using even more advanced machine learning or integrating user behavior signals, to further strengthen multi-level spam protection in the evolving threat landscape.

4. Conclusions

Spam and phishing threats continue to evolve in sophistication, but a multi-level spam protection architecture provides a resilient defense by combining multiple detection techniques. In this paper, we presented a comprehensive analysis of an integrated approach that leverages rule-based filtering, blacklist-based filtering, and machine learning classifiers in a cohesive system. Focusing on corporate email environments, we discussed how each layer addresses different facets of the spam problem – from network-level reputation to content heuristics to statistical pattern recognition – and demonstrated through recent research data how the synergy of these layers yields superior performance.

The proposed architecture achieves high spam detection rates (on the order of 99%) while maintaining very low false positives, a balance that single-method filters struggle to attain. By stopping the majority of spam at the network perimeter via DNSBL blacklists and protocol checks, the system conserves resources and immediately blocks known bad senders (e.g., one study noted blacklists can eliminate 50–70% of spam with virtually no false hits). The rule-based layer then filters out emails exhibiting known spam signatures or violating organizational policies – this adds an interpretable, customizable safety net that can catch trickier spam that made it past the blacklist (such as a spam email from a previously unseen IP that nonetheless contains suspicious phrases or formats). Finally, the machine learning layer provides a dynamic and data-driven analysis that can identify spam by statistical consistency (or inconsistency) with legitimate mail. As evidenced by multiple studies between 2020–2024, modern ML models (even relatively basic ones like logistic regression or Naïve Bayes) are highly effective, often detecting >95% of spam on their own. In our multi-layer system, the ML layer serves as the last line of defense, significantly reducing any residual misses by the earlier layers.

We also highlighted the architecture's efficiency. The layered filtering does not introduce prohibitive latency – each email undergoes a series of swift checks (milliseconds each), and the majority of emails get filtered out by the first one or two layers, meaning the expensive operations are only performed on a fraction of messages. Empirical analysis suggests the combined pipeline can easily scale to enterprise email volumes with proper optimization and parallelism, as the per-email processing time remains in the order of a few tens of milliseconds in the worst case (Figure 3). This demonstrates that enhancing detection capability does not necessarily come at the cost of performance or throughput, especially when the system is designed to short-circuit on detection.

The advantages of the multi-level architecture are clear: improved accuracy, reduced risk of falling for new spam tricks, flexibility in tuning, and meaningful alerts/explanations for administrators. However, deploying such a system in practice requires careful maintenance. We must continuously update the blacklist sources (subscribing to reputable threat intelligence feeds and ensuring timely DNSBL queries), update and refine rulesets (perhaps aided by automated analysis of new spam trends), and retrain ML models regularly with fresh data. The corporate environment also offers the opportunity to incorporate user feedback loops – for example, if users mark a spam email that evaded filters, that email can be analyzed to generate a new rule or be included in the next ML training batch, thereby plugging the gap. Over time, the multi-layer filter «learns» both from global spam evolution and the specific organization's email patterns, becoming more effective.

This architecture is also extensible. Future enhancements could include adding more layers or parallel filters, such as an anomaly detection layer that looks for unusual email patterns (e.g., a sudden spike in emails from a normally quiet partner might indicate a compromised account sending spam). Advanced machine learning techniques like deep learning (using transformer-based language models) could potentially replace or augment the current ML layer to catch more subtle social engineering emails that rely on context and language nuance (recent research using BERT-based classifiers for phishing detection shows promise in catching things that keyword-based rules might miss). Additionally, image analysis can be integrated to detect spam images or QR codes that spammers use to bypass text filters. Another emerging approach is to incorporate network-level traffic patterns (e.g., noticing many emails with similar content coming from different sources) which could be a sign of a dispersed spam campaign – this could be considered a higher-level analysis layer.

One more aspect worth considering is integration with user-level filtering. Our proposed system functions at the mail server/gateway level (before emails reach inboxes). Some enterprise setups also empower users with mailbox-level spam filters or plugins that learn individual preferences. A multi-level architecture can coexist with such systems; in fact, if something passes the server's multi-layer filter, a user's personal filter (like a Bayesian filter in their mail client) could serve as a last personalized layer. In practice, though, a well-implemented server-side multi-layer filter aims to eliminate the need for per-user filtering (except perhaps a user's personal safe-senders list).

In conclusion, the multi-level spam protection architecture presented here represents a state-of-the-art design for enterprise spam defense, synthesizing the best of deterministic rules, collaborative blacklists, and adaptive machine learning. Our analysis, grounded in contemporary research findings, confirms that this integrated approach significantly outperforms single-method strategies in both detection coverage and reliability. By deploying such an architecture, organizations can dramatically reduce the volume of spam and phishing emails that reach their employees, thereby decreasing the likelihood of successful attacks (like phishing-induced breaches or malware infections) and alleviating the productivity drain caused by junk email. As spam tactics

continue to evolve, so too can this architecture: it provides a flexible framework where new detection modules can be slotted in and layers adjusted to address new threats, whether it's new forms of phishing, social engineering, or abuse of emerging communication protocols. Ultimately, a multi-layered defense, much like defense in depth in network security, offers the robustness and adaptability needed to keep pace with adversaries in the spam arms race.

References

[1] Statista. (2023). Share of global e-mail traffic classified as spam from 2014 to 2023. *Retrieved from Statista database*

[2] AgainstData. (2025). Email Spam Statistics to Know in 2025. (Blog post with compiled statistics on spam volume and impact)

[3] Securelist (Kaspersky Lab). (2024). *Spam and Phishing Report 2023 (Annual report)*

[4] FBI Internet Crime Complaint Center (IC3). (2023). 2022 Internet Crime Report. *FBI Press Release*

[5] Britton, M. (2023). 2022 FBI IC3 Report Shows $2.7 Billion in Losses from Business Email Compromise. *Abnormal Security Blog*

[6] Iverson, A. (2007). SORBS: Accuracy Rates and False Positives. *DNSBL Resource (dnsbl.com blog)*

[7] Spamhaus. (2020). The Domain Block List (DBL). Release Notes. *Vendor technical note*

[8] Apache SpamAssassin. (2021). SpamAssassin Overview and False Positives. *Apache Wiki*

[9] Ahmed, N. et al. (2022). Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges. *Security and Communication Networks, 1862888*

[10] Kontsevaia, L., & Tutubalina, E. (2021). Effective Spam Detection with Machine Learning. *Sciendo Journal*

[11] Ghosh, A., & Senthilrajan, A. (2023). Comparison of machine learning techniques for spam detection. *Multimedia Tools and Applications, 82, 28651–28667*

[12] Ouyang, T., Ray, S., Allman, M., & Rabinovich, M. (2014). A Large-Scale Empirical Analysis of Email Spam Detection through Network Characteristics in a Stand-Alone Enterprise. *Computer Networks, 59, 101–121*

[13] Ghogare, P. P., Dawoodi, H. H., & Patil, M. P. (2023). Enhancing Spam Email Classification Using Effective Preprocessing Strategies and Optimal Machine Learning Algorithms. (Preprint)

[14] Nazgol, T. et al. (2020). Forecasting Malicious Email with Multiple Protection Layers. *IEEE Conference Paper*

[15] Karyawati, A. E. et al. (2023). A Comparison of Different Kernel Functions of SVM for Spam Detection. *Jurnal Ilmu Komputer, 8(2), 91–97*

[16] SpamLaws. (2023). Spam Statistics and Facts. (Online resource)

[17] Respective Product Documentation (2024). Enterprise Email Security Gateway Guides. (e.g., Cisco, Proofpoint, Microsoft EOP)

[18] Hemalatha, M. et al. (2022). E-mail Spam Detection. International Journal of Computer Science and Mobile Computing, 11(1), pp. 36–44.

[19] Securelist (Kaspersky Lab). (2021). Spam and Phishing in Q4 2021

Спамнан қорғаудың көп деңгейлі архитектурасы: ережелерді, қара тізімдерді және машиналық оқыту әдістерін біріктіру

О. Кадыров*, А. Батыргалиев

Satbayev University, Алматы, Қазақстан

**Корреспонденция үшін автор: omarkadirov70@gmail.com*

Аңдатпа. Электрондық пошта спамы Әлемдік пошта трафигінің жартысына жуығын құрайтын корпоративтік байланыстарға айтарлықтай қауіп төндіруді жалғастыруда. Спам, әсіресе фишингтік хаттар, пошта жәшіктерін бітеп қана қоймайды, сонымен қатар елеулі қаржылық шығындарға әкеледі: мысалы, тек 2022 жылы Іскери электрондық поштаны бұзу алаяқтығы 2,7 миллиард АҚШ долларын құрады. Бұл мақалада үш қосымша тәсілді біріктіретін көп деңгейлі спамға қарсы архитектура ұсынылады: ережеге негізделген сүзу, қара тізімге негізделген сүзу және машиналық оқытуға негізделген жіктеу (ML). Біз әр деңгейдің жұмысын егжей-тегжейлі сипаттаймыз және олардың корпоративті ортадағы күшті және әлсіз жақтарын талдаймыз, мұнда жоғары дәлдік пен жалған позитивтердің минималды саны өте маңызды. Ұсынылған архитектура Протокол деңгейіндегі қорғаныс құралдарын (мысалы, DNS негізіндегі қара тізімдер), мазмұнға негізделген эвристиканы (ережелер жүйесі) және ML классификаторларын (Naïve Bayes, Decision Trees, Logistic Regression) көп деңгейлі "терең қорғаныс"моделіне біріктіреді. Біз бұл интеграцияланған тәсіл жалған позитивтердің санын бақылау кезінде спамды анықтау көрсеткіштерін айтарлықтай жақсарта алатынын көрсету үшін соңғы зерттеулерді (2020-2024) пайдалана отырып салыстырмалы бағалау жүргіземіз. Әдебиеттерден нақты статистика мен эксперименттік нәтижелер, соның ішінде өнімділік көрсеткіштері (дәлдік, дәлдік, кері байланыс, F1-score) және әр әдіс үшін жұмыс уақыты ұсынылған. Көп қабатты жүйе кез-келген жеке техникамен салыстырғанда жоғары өнімділікке қол жеткізеді (көбінесе жалған позитивтердің төмен деңгейінде 98-99% анықтау). Бұл жұмыс корпоративтік спамнан қорғаудың сенімді құрылымын сипаттайтын және адаптивті, көп қырлы спамды сүзу саласындағы зерттеулердің болашақ бағыттарын қамтитын тәжірибешілер мен зерттеушілер үшін жан-жақты және өзекті талдауды ұсынады.

Негізгі сөздер: *спам, спамды сүзу, электрондық пошта қауіпсіздігі, кәсіпорын желілері, көп деңгейлі сүзу, ережеге негізделген сүзу, DNS негізіндегі қара тізімдер, DNSBL, URIBL, Машиналық оқыту, аңғал Байес, шешім ағаштары, логистикалық регрессия, пошта серверлері, спамды анықтау, фишингтің алдын алу, мазмұнды сүзу, IP беделі, сүзу қосулы хаттама деңгейі, Электрондық поштаның жіктелуі, спамға қарсы архитектура*.

Многоуровневая архитектура защиты от спама: интеграция правил, черных списков и методов машинного обучения

О. Кадыров*, А. Батыргалиев
Satbayev University, Алматы, Казахстан
**Автор для корреспонденции: omarkadirov70@gmail.com*

Аннотация. Спам в электронной почте продолжает представлять значительную угрозу для корпоративных коммуникаций, составляя почти половину мирового почтового трафика. Спам, особенно фишинговые письма, не только засоряет почтовые ящики, но и приводит к серьезным финансовым потерям: например, только на мошенничество с компрометацией деловой электронной почты в 2022 году пришлось 2,7 млрд долларов США. В данной статье предлагается многоуровневая архитектура защиты от спама, объединяющая три взаимодополняющих подхода: фильтрацию на основе правил, фильтрацию на основе черных списков и классификацию на основе машинного обучения (ML). Мы подробно описываем функционирование каждого уровня и анализируем их сильные и слабые стороны в корпоративной среде, где высокая точность и минимальное количество ложных срабатываний имеют первостепенное значение. Предлагаемая архитектура объединяет средства защиты на уровне протоколов (например, черные списки на базе DNS), эвристику на базе контента (системы правил) и классификаторы ML (Naïve Bayes, Decision Trees, Logistic Regression) в многоуровневую модель «защита в глубину». Мы проводим сравнительную оценку с использованием последних исследований (2020-2024 гг.), чтобы продемонстрировать, что этот интегрированный подход может значительно улучшить показатели обнаружения спама, контролируя при этом количество ложных срабатываний. Представлена реальная статистика и экспериментальные результаты из литературы, включая показатели производительности (точность, прецизионность, отзыв, F1-score) и время выполнения для каждого метода. Многослойная система достигает превосходной производительности (часто 98-99 % обнаружения при низком уровне ложных срабатываний) по сравнению с любой отдельной методикой. Данная работа предлагает всесторонний и актуальный анализ для практиков и исследователей, описывая надежную структуру защиты от корпоративного спама и освещая будущие направления исследований в области адаптивной, многогранной фильтрации спама.

Ключевые слова: *спам, фильтрация спама, безопасность электронной почты, корпоративные сети, многоуровневая фильтрация, фильтрация на основе правил, черные списки на основе DNS, DNSBL, URIBL, машинное обучение, наивный Байес, деревья решений, логистическая регрессия, почтовые серверы, обнаружение спама, предотвращение фишинга, контентная фильтрация, IP-репутация, фильтрация на уровне протокола, классификация электронной почты, архитектура антиспама*.

Received: 03 April 2025
Accepted: 15 June 2025
Available online: 30 June 2025