algorithms

MDPI

Review

Foundational Algorithms for Modern Cybersecurity: A Unified Review on Defensive Computation in Adversarial Environments

Paul A. Gagniuc

Faculty of Engineering in Foreign Languages, National University of Science and Technology Politehnica Bucharest, RO-060042 Bucharest, Romania; paul.gagniuc@upb.ro

Abstract

Cyber defense has evolved into an algorithmically intensive discipline where mathematical rigor and adaptive computation underpin the robustness and continuity of digital infrastructures. This review consolidates the algorithmic spectrum that supports modern cyber defense, from cryptographic primitives that ensure confidentiality and integrity to behavioral intelligence algorithms that provide predictive security. Classical symmetric and asymmetric schemes such as AES, ChaCha20, RSA, and ECC define the computational backbone of confidentiality and authentication in current systems. Intrusion and anomaly detection mechanisms range from deterministic pattern matchers exemplified by Aho-Corasick and Boyer-Moore to probabilistic inference models such as Markov Chains and HMMs, as well as deep architectures such as CNNs, RNNs, and Autoencoders. Malware forensics combines graph theory, entropy metrics, and symbolic reasoning into a unified diagnostic framework, while network defense employs graph-theoretic algorithms for routing, flow control, and intrusion propagation. Behavioral paradigms such as reinforcement learning, evolutionary computation, and swarm intelligence transform cyber defense from reactive automation to adaptive cognition. Hybrid architectures now merge deterministic computation with distributed learning and explainable inference to create systems that act, reason, and adapt. This review identifies and contextualizes over 50 foundational algorithms, ranging from AES and RSA to LSTMs, graph-based models, and post-quantum cryptography, and redefines them not as passive utilities, but as the cognitive genome of cyber defense: entities that shape, sustain, and evolve resilience within adversarial environments.

Keywords: cyber defense; cryptographic algorithms; anomaly detection; malware analysis; graph theory; reinforcement learning; swarm intelligence; hybrid architectures; post-quantum cryptography; explainable AI

check for updates

Academic Editor: Francesco Bergadano

Received: 11 October 2025
Revised: 30 October 2025
Accepted: 3 November 2025
Published: 7 November 2025

Citation: Gagniuc, P.A. Foundational Algorithms for Modern Cybersecurity: A Unified Review on Defensive Computation in Adversarial Environments. *Algorithms* **2025**, *18*, 709. https://doi.org/10.3390/a18110709

1. Introduction

The increasing sophistication of cyberattacks reveals a deep evolutionary trajectory from rule-based intrusions to intelligent, adaptive adversaries that exploit algorithmic weaknesses at both software and hardware levels [1]. In this adversarial landscape, algorithms no longer function as mere computational tools but as the core layer upon which digital defense architectures are engineered [2]. Cryptographic primitives define the mathematical boundary of confidentiality and integrity [3], while identification and behavioral analysis algorithms constitute the cognitive layer of defense, capable to detect deviations, learn from adversarial patterns, and predict unseen threats [4]. The central research question addressed in this review is: Which foundational algorithms constitute the computational core of modern cyber defense, and how do they maintain effective performance under adversarial conditions? The purpose of this review is to delineate and classify the essential algorithms that underlie modern cyber defense systems, organized by operational logic and strategic role, from deterministic encryption schemes to stochastic models of behavioral inference (Figure 1) [5]. This taxonomic approach reveals how distinct algorithmic paradigms coalesce into an integrated defense continuum, where mathematical abstraction meets adversarial pragmatism (Figure 1) [6]. Here, algorithm selection was guided by three principal criteria: historical relevance, computational efficiency, and stability under adversarial perturbation.

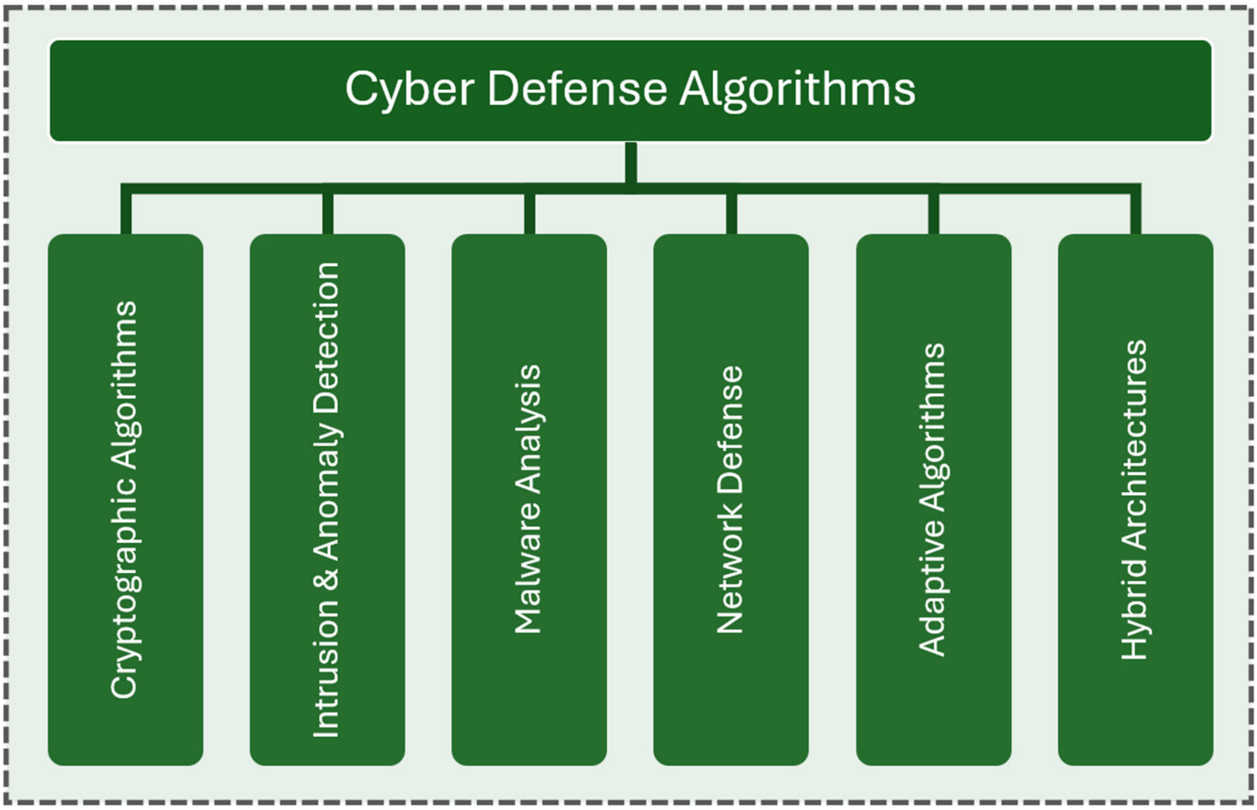

Figure 1. Taxonomy of Cyber Defense Algorithms. The figure illustrates the main categories of algorithmic paradigms used in cybersecurity. Each branch represents a distinct domain: cryptographic primitives, detection mechanisms, reverse engineering methods, graph-theoretic defense strategies, behavioral and adaptive techniques, and hybrid architectures. These six pillars form the structural basis of algorithmic design against adversarial threats (For classification please refer to Appendix A).

Historical relevance ensures inclusion of algorithms that have shaped cryptographic and detection theory over decades, such as RSA and Hidden Markov Models [7]. Efficiency encompasses asymptotic complexity (i.e., the rate at which time or space requirements increase with input size), energy footprint, and feasibility of real-time deployment in constrained environments [8]. Also, the adaptive strength of algorithms is critical when confronted with quantum threats, adversarial learning, and polymorphic malware [9]. Together, these dimensions provide a coherent framework to evaluate not only what algorithms are, but how they endure under evolving attack surfaces [10]. The review proceeds as follows: Section 2 analyzes cryptographic algorithms; Section 3 examines intrusion and anomaly detection; Section 4 addresses malware forensics; Section 5 focuses on network defense; Section 6 presents behavioral and adaptive algorithms; Section 7 explores hybrid architectures; and Section 8 concludes with integrative remarks.

Review Methodology

The review followed PRISMA-inspired guidelines adapted for algorithmic and computational studies. Literature was retrieved from IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and arXiv within the 2018–2025 interval using combined keyword sets such as "algorithm AND cybersecurity," "cryptography," "intrusion detection," and "forensics." Inclusion criteria emphasized peer-reviewed works addressing algorithmic robustness, adversarial resistance, or hybrid cyber-defense architectures. After duplicate removal and abstract screening, 96 studies met the inclusion thresholds based on technical depth, methodological soundness, and relevance to resilient computation. Foundational algorithmic works predating 2018 (e.g., AES, HMM, SVM) were retained where historically essential to trace theoretical continuity. This structured process ensures both historical completeness and representation of the most recent advances, and thus provides a transparent basis for the unified taxonomy presented in the subsequent sections.

2. Cryptographic Algorithms: Confidentiality and Integrity

At the heart of cyber defense lies the duality of symmetric and asymmetric encryption, namely the mathematical foundations of confidentiality and authentication [11]. Symmetric algorithms, represented by the Advanced Encryption Standard (AES), employ substitution-permutation networks that blend algebraic diffusion with nonlinear confusion to resist linear and differential cryptanalysis [12]. The Rijndael structure of AES, defined over the finite field GF(2^8) (i.e., Galois Field), achieves computational elegance and resists side-channel exploitation; it remains the global standard for hardware-accelerated confidentiality (Figure 2) [13].

Modern cryptographic implementations employ masking, blinding, and randomization to reduce information leakage from power, timing, or electromagnetic side channels. Boolean and arithmetic masking randomize intermediate computations to decorrelate secret-dependent transitions, while exponent blinding and randomized projective coordinates in RSA and ECC counteract differential power analysis (DPA). Such countermeasures complement algorithmic design, and ensure robustness not only at the mathematical but also at the physical level of execution. In contrast, ChaCha20, derived from the Salsa family, reimagines stream ciphers through ARX (Addition-Rotation-XOR) primitives [14]. Asymmetric encryption extends this protection from secrecy to identity. RSA, built upon the hardness of integer factorization, formalized the first scalable public-key paradigm and remains foundational in digital signatures and key exchange protocols [15]. However, its reliance on modular exponentiation over large primes imposes computational inertia, which prompted the emergence of Elliptic Curve Cryptography (ECC), that achieves equivalent security through algebraic operations on elliptic curves over finite fields [16]. The Elliptic Curve Diffie-Hellman (ECDH) protocol further refines key agreement through conversion of the intractability of the discrete logarithm problem into practical efficiency and through compression of cryptographic strength into minimal bit length (Figure 2) [17].

Hashing algorithms close the trinity of confidentiality, integrity, and authentication. The SHA-3 family, built on the Keccak sponge construction, introduced a permutation-based paradigm resistant to length-extension and collision vulnerabilities that compromised earlier standards [18]. BLAKE3, a cryptographic synthesis of the BLAKE2 lineage, integrates tree hashing with SIMD parallelism to achieve linear scalability on modern multi-core systems [19]. Argon2, the winner of the Password Hashing Competition, extends beyond pure hashing through inclusion of memory-hard functions designed to neutralize GPU and ASIC acceleration in brute-force attacks [20]. The post-quantum horizon redefines the boundaries of secure computation. Lattice-based schemes such as CRYSTALS-Dilithium exploit the hardness of the Learning With Errors (LWE) problem to construct efficient and quantum-resistant digital signatures [21]. Similarly, hash-based families like SPHINCS+ derive their security guarantees from the one-wayness of hash functions, trading key reuse for provable security under quantum adversaries [22]. These constructions foreshadow a paradigmatic shift, from factoring and logarithmic hardness to geometric and combinatorial entropy [23].

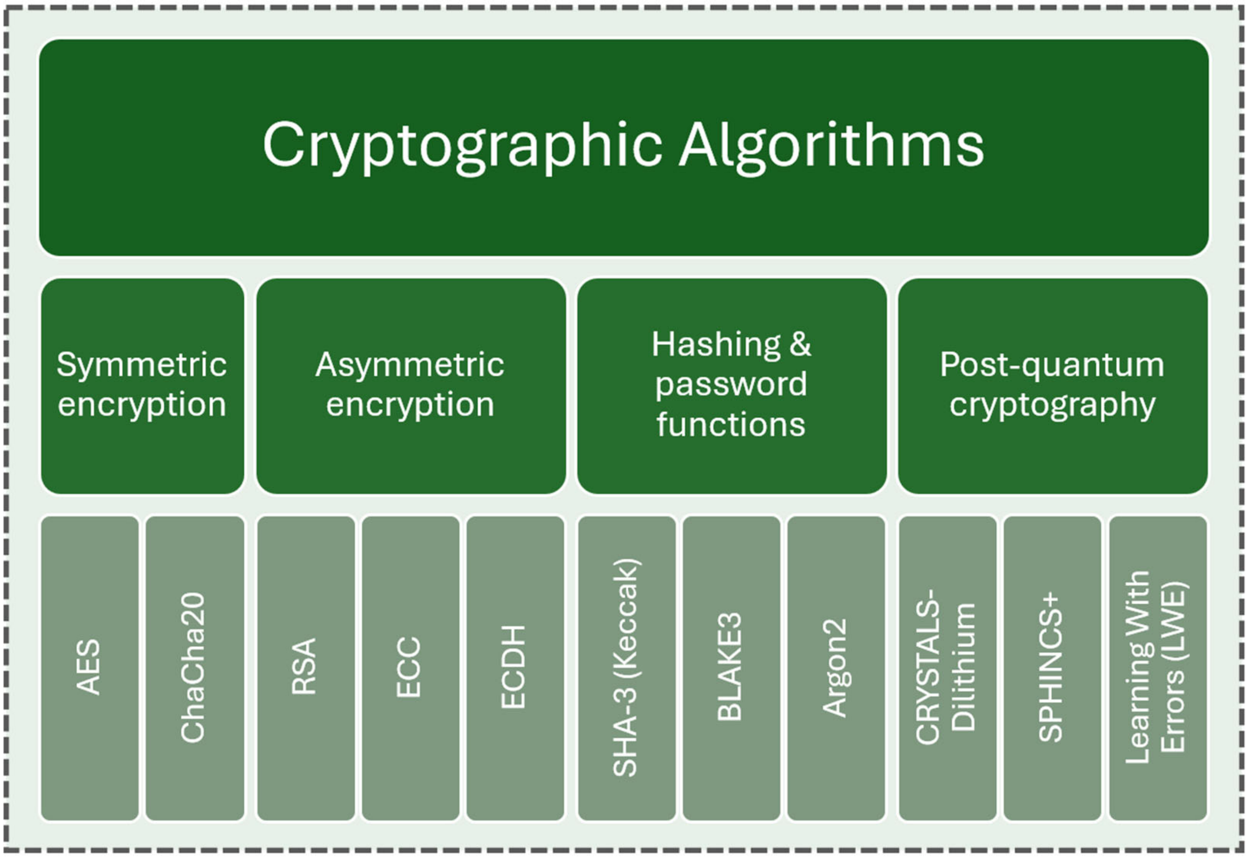

Figure 2. Taxonomy of Cryptographic Algorithms. This figure categorizes cryptographic algorithms into four major groups: symmetric encryption (e.g., AES, ChaCha20), asymmetric encryption (e.g., RSA, ECC, ECDH), hashing and password functions (e.g., SHA-3, BLAKE3, Argon2), and post-quantum cryptographic primitives (e.g., CRYSTALS-Dilithium, SPHINCS+, Learning With Errors). Each class addresses a distinct security requirement, confidentiality, integrity, authentication, and quantum resistance, within modern cyber defense systems.

3. Algorithms for Intrusion and Anomaly Detection

Intrusion detection represents the cognitive frontier of cyber defense, where algorithms must discern malicious deviation amid stochastic digital noise (Figure 3) [24]. The earliest line of defense, the signature-based detection, operationalizes linguistic matching through deterministic automata. The Aho-Corasick algorithm constructs finite-state machines for multi-pattern recognition with linear-time guarantees, making it the canonical model for deep packet inspection and antivirus scanning [25]. Boyer-Moore complements this approach by reversing the search paradigm: instead of checking every character sequentially, it exploits heuristic skipping through bad-character and good-suffix shifts, which in turn enables sublinear average-case complexity and secures a central role in string-based intrusion engines for decades [26]. Yet, static signatures fail under high levels of polymorphism and zero-day variability, and push for the rise of statistical and probabilistic paradigms. The Naive Bayes classifier, despite reliance on a simplification of conditional independence, remains one of the earliest and most interpretable probabilistic models for network intrusion detection, where feature distributions approximate conditional likelihoods of attack classes [27].

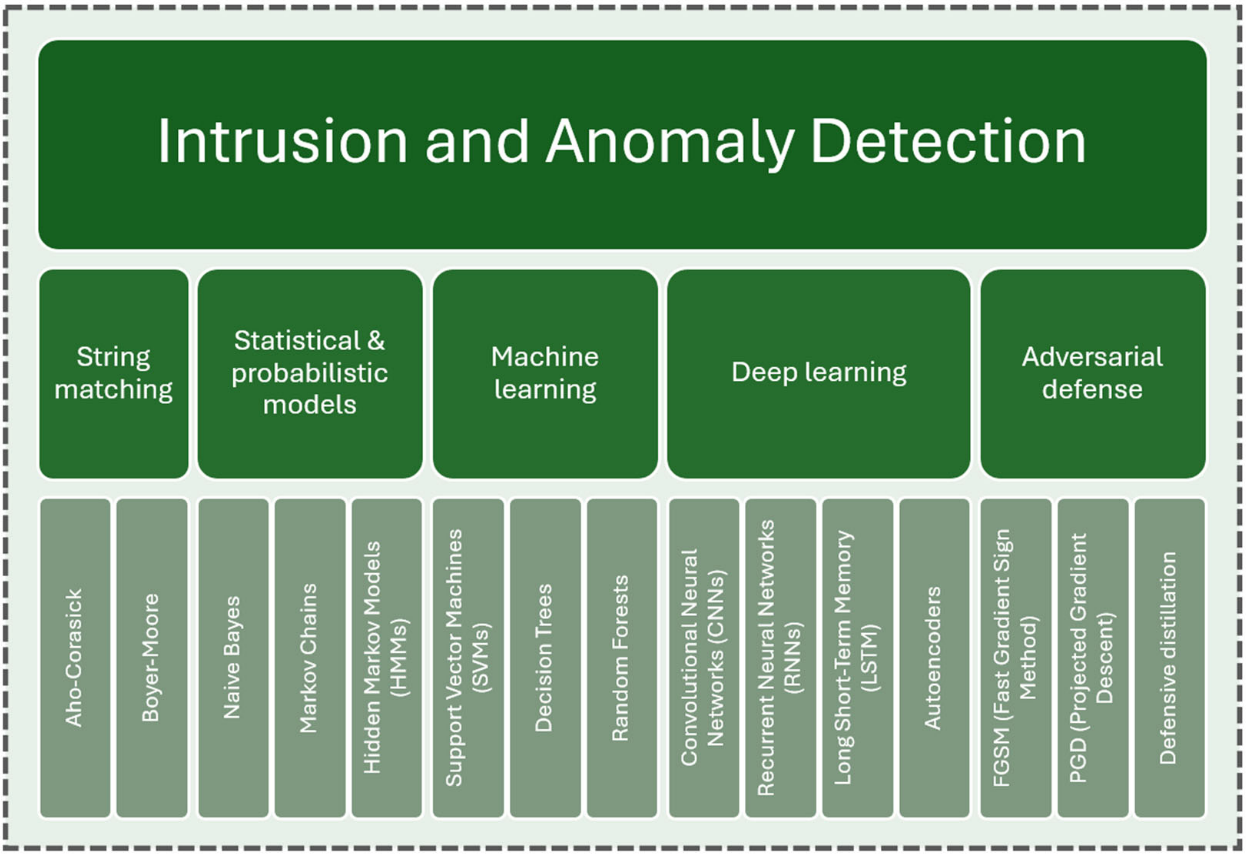
Intrusion and Anomaly Detection
String matching
Statistical & probabilistic models
Machine learning
Deep learning
Adversarial defense
Aho-Corasick
Boyer-Moore
Naive Bayes
Markov Chains
Hidden Markov Models (HMMs)
Support Vector Machines (SVMs)
Decision Trees
Random Forests
Convolutional Neural Networks (CNNs)
Recurrent Neural Networks (RNNs)
Long Short-Term Memory (LSTM)
Autoencoders
FGSM (Fast Gradient Sign Method)
PGD (Projected Gradient Descent)
Defensive distillation

Figure 3. Taxonomy of Intrusion and Anomaly Detection Algorithms. This figure outlines major algorithmic families used in intrusion and anomaly detection. It includes classical string-matching techniques (e.g., Aho-Corasick, Boyer-Moore), statistical and probabilistic models (e.g., Naive Bayes, Markov Chains, HMMs), traditional machine learning classifiers (e.g., SVMs, Decision Trees, Random Forests), deep learning models (e.g., CNNs, RNNs, LSTM, Autoencoders), and adversarial defense mechanisms (e.g., FGSM, PGD, defensive distillation). These methods contribute to the detection of cyber threats through behavior profiling, anomaly identification, and robustness against adversarial inputs (For classification criteria, refer to Appendix A).

Markov Chains abstract system transitions into probabilistic states, which allows deviations in transition probabilities to signal intrusions at process or network levels [28]. Hidden Markov Models (HMMs) significantly extend this formulation through incorporation of latent behavioral states (i.e., behavioral dilatations), thereby making a detection of stealthy or temporally correlated anomalies invisible to direct observation [29]. While HMMs and LSTMs perform well on stationary datasets, their accuracy drops in sparse or dynamic environments. Approaches based on continual adaptation, domain transfer, and stepwise model updates allow systems to adjust to new traffic profiles and concept shifts, which helps maintain system reliability.

Machine learning extended these probabilistic frameworks into nonlinear discriminative landscapes. Support Vector Machines introduced the concept of maximizing margin hyperplanes (i.e., boundaries that best divide different types of data) in high-dimensional feature spaces, creating robust decision boundaries even with scarce labeled intrusion data [30]. Decision Trees and their ensemble extensions, such as Random Forests, provide hierarchical feature partitioning that balances interpretability with variance reduction, which further allows for a real-time deployment in Security Information and Event Management (SIEM) systems [31]. Deep learning subsequently redefined anomaly detection through representation learning. Convolutional Neural Networks (CNNs) capture spatial correlations in traffic matrices and malware binaries, and convert packet flows into structured tensors amenable to gradient-based optimization [32]. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) architectures, exploit temporal dependencies in event sequences, which enables predictive defense against multi-stage attacks [33]. Autoencoders, trained to reconstruct normal behavior, achieve unsupervised detection of anomalies through reconstruction error, that acts as adaptive filters in high-dimensional behavioral space [34]. Furthermore, these models (collectively) transform cyber defense from reactive filtering to anticipatory cognition, where algorithms evolve from passive detectors into autonomous interpreters of digital intent [35]. Adversarial manipulation of feature spaces poses a critical risk to machine-learning-based detectors. Gradient-based attacks such as FGSM (Fast Gradient Sign Method) and PGD (Projected Gradient Descent) often deceive classifiers while outputs appear normal. Methods such as defensive distillation, adversarial example defenses, and attention-based interpretation provide partial protection. However, no current approach offers complete solutions against adaptive adversaries.

4. Algorithms in Malware Analysis and Reverse Engineering

Malware analysis represents the algorithmic archeology of digital pathogens, where defense mechanisms dissect, abstract, and reconstruct adversarial intent through computational introspection (Figure 4) [36]. Graph-based algorithms dominate this forensic layer through conversion of executable binaries into topological entities. Control Flow Graphs (CFGs) formalize program execution as directed graphs where nodes correspond to basic blocks and edges encode transfer of control, and thereby enables the detection of obfuscation and code injection through structural deviation analysis [37].

Data Flow Analysis complements this paradigm by modeling the propagation of variables across instruction sets, thus allowing for a reconstruction of hidden dependencies and the identification of tainted data paths (i.e., code paths where external data can corrupt or control internal logic) that betray malicious payloads [38]. These approaches, grounded in compiler theory, allow the algorithmic fingerprinting of malware families by the invariance of their structural motifs (much like in the field of Bioinformatics), even under heavy polymorphic transformations [39]. Entropy-based detection extends this structural analysis into the statistical domain. The quantification of byte-level randomness through Shannon entropy allows analysts to distinguish encrypted or packed sections from normal code regions, as high-entropy subsequences often signify compressed or self-modifying data [40]. String analysis algorithms use pattern extraction and n-gram frequency distributions to identify polymorphic and metamorphic variants that evade static signatures, which in turn allow cross-sample clusters to form based on lexical similarity metrics (i.e., quantitative measures of textual resemblance between code fragments or strings) [41]. Entropy and string-based approaches together operationalize information theory as an adversarial microscope, that is able to show concealed semantics beneath algorithmic camouflage [42].

Symbolic execution represents the analytical peak of reverse engineering, where program paths are explored through symbolic inputs rather than concrete execution traces [43]. Constraint solvers such as Z3 and STP (i.e., automated theorem provers that solve logical constraints derived from program conditions) convert program predicates into logical formulas, which permits automatic path exploration and reveals hidden conditions that trigger malicious behavior [44]. SAT (Boolean satisfiability) and SMT (Satisfiability Modulo Theories) solvers provide the computational substrate for this reasoning and convert code deobfuscation into a logical inference problem where complexity meets deductive precision [45]. These algorithmic frameworks have proven indispensable in decoding obfuscated binaries and for the generation of decryption stubs in modern malware families such as Conficker and Emotet [46]. Thus, the fusion of graph theory, entropy analytics, and symbolic logic defines the algorithmic triad for contemporary malware forensics [47].

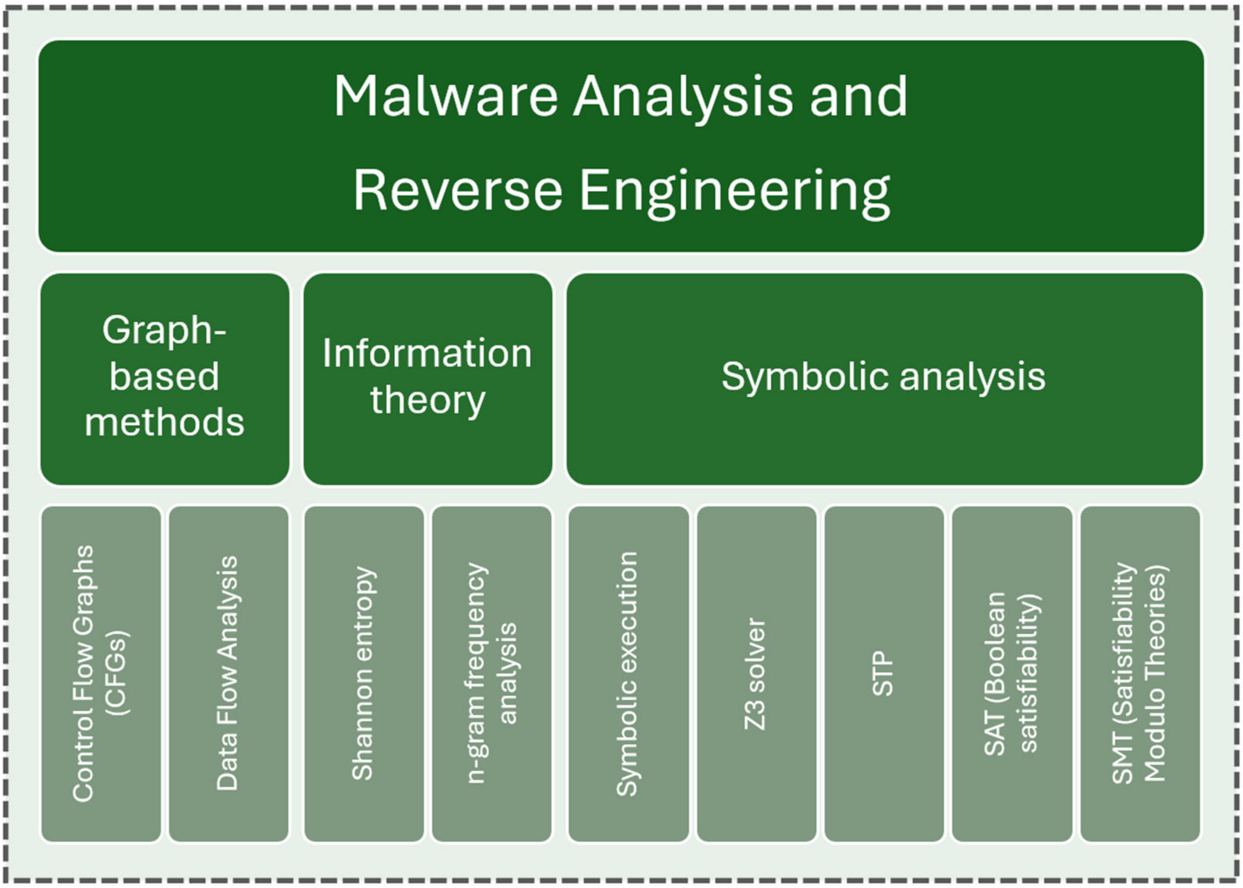
Malware Analysis and Reverse Engineering
Graph-based methods
Information theory
Symbolic analysis
Control Flow Graphs (CFGs)
Data Flow Analysis
Shannon entropy
n-gram frequency analysis
Symbolic execution
Z3 solver
STP
SAT (Boolean satisfiability)
SMT (Satisfiability Modulo Theories)

Figure 4. Taxonomy of Algorithms for Malware Analysis and Reverse Engineering. This figure organizes core techniques used in dissecting and understanding malware behavior. Graph-based methods such as Control Flow Graphs (CFGs) and Data Flow Analysis facilitate structural code analysis. Information-theoretic approaches, including Shannon entropy and n-gram frequency analysis, quantify statistical irregularities in binaries. Symbolic analysis encompasses logic-based reasoning frameworks like symbolic execution and constraint solvers (e.g., Z3, STP), including satisfiability solvers such as SAT and SMT, which enable exploration of execution paths and formal verification of malware behavior.

5. Network Defense and Graph-Theoretic Algorithms

Cyber defense within networked infrastructures is, at its core, a problem of graph optimization under adversarial uncertainty (Figure 5) [48]. Every packet route, session flow, or peer connection maps naturally to a graph structure whose topology encodes both robustness and vulnerability [49]. Shortest path algorithms, such as Dijkstra and Floyd-Warshall, originally conceived for routing optimization, have evolved into instruments of security reasoning, that are able to quantify network exposure and computation of minimal-attack surfaces across dynamic infrastructures [50]. In secure routing, the adaptation of the Dijkstra algorithm under trust-weighted metrics allows the identification of least-risk paths, while the Floyd-Warshall all-pairs formulation provides global visibility for the detection of potential pivot chains in lateral movement analysis (i.e., the examination of how an attacker expands access within a compromised network to reach additional systems or data) [51]. These algorithms thus shift from transportation logic to defensive analytics, where "distance" becomes a multidimensional measure of latency, trust, and compromise probability [52]. *Flow algorithms* extend this paradigm by modeling the network as a fluid system whose capacity constraints mirror the limits of intrusion propagation. The Ford-Fulkerson method, through iterative augmenting paths, captures the maximal data rates between nodes, which allows analysts to simulate denial-of-service dynamics and identify bottlenecks vulnerable to saturation [53]. The Edmonds-Karp refinement, with its polynomial-time bound via breadth-first traversal (i.e., a graph exploration method that visits all neighboring nodes at each depth level before proceeding to the next), facilitates real-time anomaly detection in software-defined networks through comparison of expected versus observed flow distributions [54]. These flow models translate adversarial pressure into measurable network tension, where algorithmic imbalance signifies the presence of stealthy congestion attacks or malicious routing loops [55].

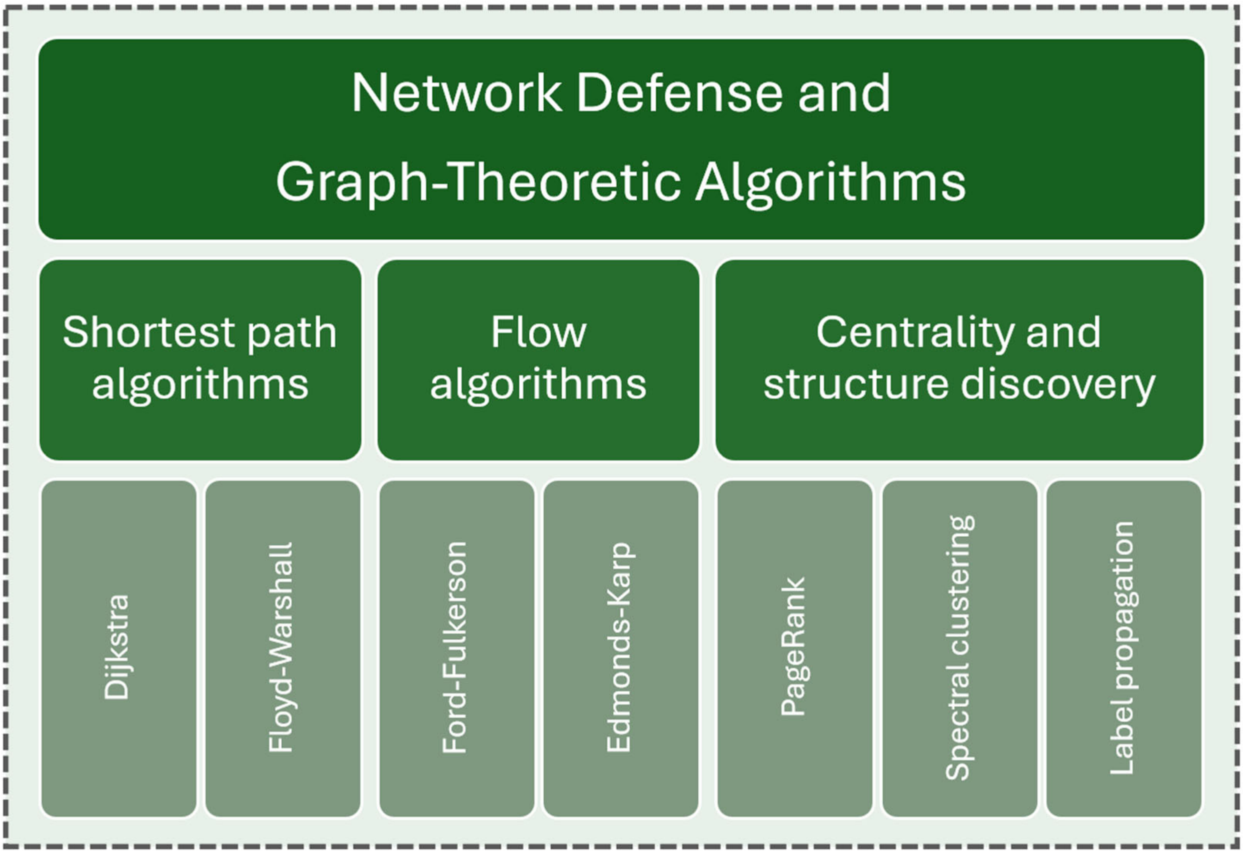

Figure 5. Taxonomy of Network Defense and Graph-Theoretic Algorithms. This figure categorizes key algorithms used to defend cyber infrastructure through analysis of network topology and data flows. Shortest path algorithms such as Dijkstra and Floyd-Warshall support optimal routing and threat tracing. Flow algorithms like Ford-Fulkerson and Edmonds-Karp aid in the identification of bottlenecks and intrusion pathways. Centrality and structure discovery methods, PageRank, spectral cluster analysis, and label propagation (i.e., spread of node class names via graph links), facilitate threat localization, influence analysis, and anomaly detection in dynamic or large-scale networked environments.

At a higher level of abstraction, graph-theoretic centrality and community detection algorithms, unveil the latent social architecture of cyber ecosystems. PageRank-like eigenvector methods, initially designed for web indexing, now quantify influence and persistence within botnet infrastructures, ranking nodes not by content but by propagation potential [56]. Community detection via modularity optimization, spectral clustering, or label propagation, exposes coordinated subgraphs of compromised hosts, forming algorithmic signatures of distributed attacks [57]. When combined, these approaches create a defensive epistemology (i.e., the study of knowledge) where topology reveals intent, an analytical transition from traffic to structure, and from structure to strategy [58]. In essence, network defense has evolved into a field where graph algorithms operate not merely as routing tools but as cognitive instruments to see the geometry of digital conflict [59].

6. Behavioral and Adaptive Algorithms

As cyber threats evolve beyond static signatures, defense mechanisms must internalize adaptation as a mathematical reflex rather than a programmed response (Figure 6) [60]. Reinforcement learning (RL) redefines defensive logic that transformes network protection into a sequential decision problem where algorithms maximize long-term security rewards through policy optimization [61]. In Q-learning (i.e., a reinforcement method that learns optimal actions from trial-and-error reward feedback), agents learn optimal countermeasures by mapping environmental states to action-value functions without explicit models of adversarial dynamics, that further allows for autonomous adaptation to novel attack patterns [62].

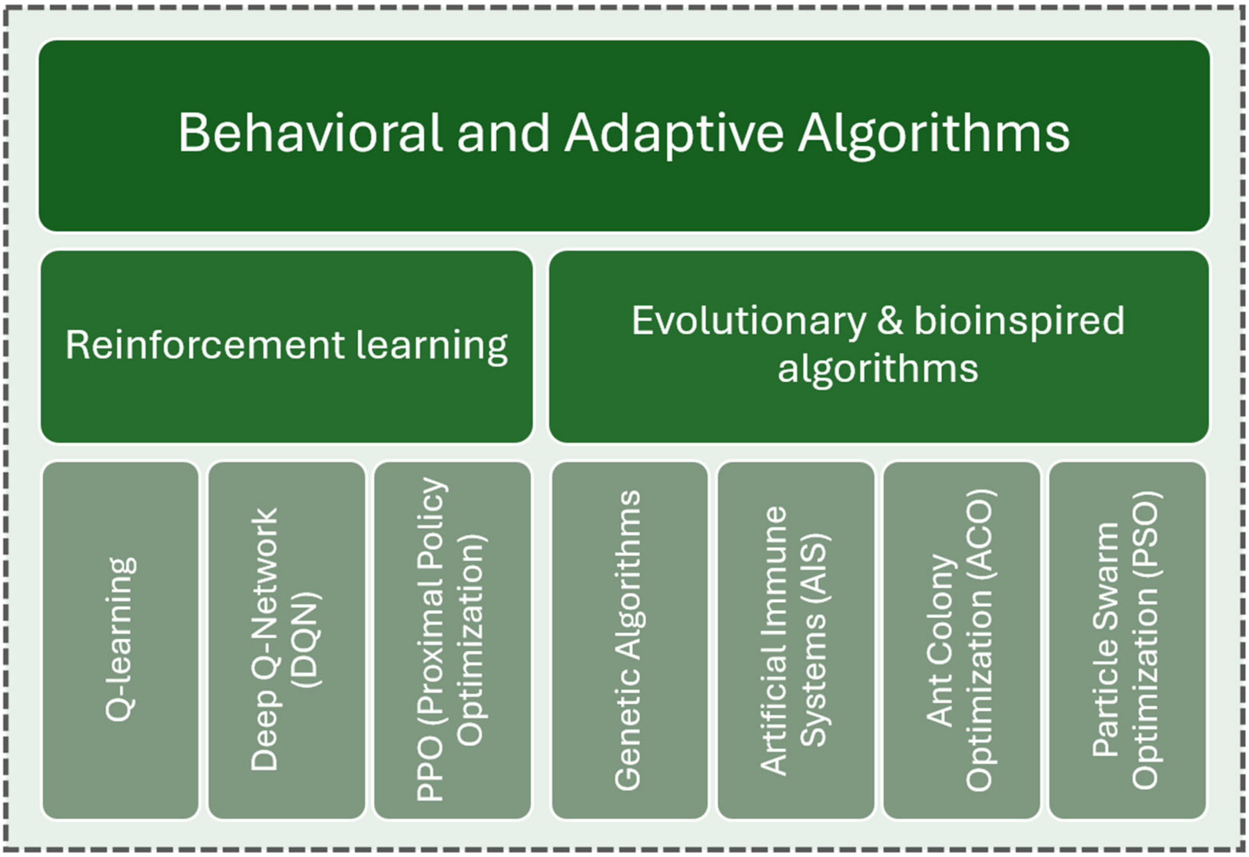
Behavioral and Adaptive Algorithms
Reinforcement learning
Evolutionary & bioinspired algorithms
Q-learning
Deep Q-Network (DQN)
PPO (Proximal Policy Optimization)
Genetic Algorithms
Artificial Immune Systems (AIS)
Ant Colony Optimization (ACO)
Particle Swarm Optimization (PSO)

Figure 6. Taxonomy of Behavioral and Adaptive Algorithms in Cyber Defense. This figure classifies algorithms that enable autonomous, adaptive, and self-improving responses to evolving cyber threats. Reinforcement learning methods, such as Q-learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO), optimize decision-making through trial-and-error interactions. Evolutionary and bioinspired algorithms, including Genetic Algorithms, Artificial Immune Systems (AIS), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO), draw from natural processes to generate robust and flexible defense strategies (For classification criteria, refer to Appendix A).

Despite their theoretical appeal, reinforcement learning models such as *Q-learning* and DQNs often suffer from poor sample efficiency and convergence instability, particularly in high-stakes defensive environments. Stabilization methods, such as prioritized experience replay, double *Q-learning*, and policy-gradient approaches like PPO, offer incremental improvements, but, remain computationally expensive for real-time defense. Deep Q-Networks (DQNs) extend this principle through deep function approximation, where convolutional layers infer latent security states from high-dimensional telemetry, that produce self-tuning intrusion containment strategies capable of anticipating multistep adversarial behavior [63]. These architectures render cyber defense not reactive but anticipatory, and embed cognition into the algorithmic fabric of security operations [64].

Evolutionary computation introduces adaptability through biological metaphor. For instance, genetic algorithms, driven by principles of selection, crossover, and mutation, evolve candidate defense configurations toward global optima, and balance detection sensitivity and false-positive suppression in dynamic environments [65]. Immune-inspired algorithms further refine this paradigm through negative selection and clonal expansion mechanisms, with emulation of self-nonself discrimination to identify anomalous patterns analogous to immune antigens in artificial immune systems (AIS) [66]. Such bioinspired models exhibit intrinsic fault tolerance and self-healing, dynamically reconfiguring defensive policies in response to system perturbations or compromise [67]. Swarm intelligence completes this adaptive triad through decentralization of decision-making across populations of lightweight agents. Moreover, Ant Colony Optimization (ACO), rooted in pheromone trail reinforcement, enables distributed path discovery in network defense, a strategy that optimizes sensor deployment or threat containment routes under incomplete information [68]. Furthermore, Particle Swarm Optimization (PSO), inspired by collective behavior of flocks, orchestrates resource allocation and parameter tuning across intrusion detection components, converging toward equilibrium between exploration and exploitation in defense space [69]. Both paradigms embody emergent intelligence, that is, a macroscopic order arising from microscopic autonomy, thus leading to a transformation of cybersecurity infrastructures into self-organizing ecosystems capable of robustness through collaboration [70]. Ultimately, behavioral and adaptive algorithms mark the transition from defensive automation to algorithmic sentience, where systems not only learn from threats but evolve in synchrony with them [71].

7. Algorithmic Integration and Hybrid Defense Architectures

The convergence of classical and intelligent algorithms within modern cyber defense systems marks a decisive transition from isolated detection to cohesive cognitive architectures (Figure 7) [72]. Security Information and Event Management (SIEM) frameworks now embody algorithmic ecosystems where deterministic rule-based mechanisms cohabit with probabilistic and neural models, which may achieve both speed and contextual inference [73]. Intrusion Detection Systems (IDS), once governed by signature and threshold logic, now operate as hybrid inference engines (i.e., systems that infer decisions from data instead of executing fixed rules). AES-encrypted telemetry streams pass through Markovian predictors and deep anomaly classifiers in tandem, that combines formal verification with adaptive learning [74]. Antivirus architectures follow similar trajectories: heuristic scanners augmented by recurrent neural filters decode behavioral entropy in binaries, and construct temporal signatures that evolve with exposure [75]. Even firewalls, historically static boundary enforcers, have become algorithmically reflexive and now deploy reinforcement-learning agents that modulate access control policies in real time based on risk gradients computed from live network graphs (i.e., mathematical structures that represent systems as nodes connected by communication links) [76]. Hybridization reaches its most transformative expression in distributed and cloud-based detection ecosystems. Cloud-native security frameworks exploit federated learning paradigms (i.e., a decentralized training approach where multiple devices or institutions collaboratively update a shared model without exchanging raw data) to share model gradients (i.e., numerical vectors that specify how model parameters must adjust to reduce prediction error), and thus preserve confidentiality while allowing global adaptation across decentralized domains [77]. Each node contributes a locally trained model, forming a collective intelligence that resists data corruption and privacy leakage [78]. These distributed architectures establish an equilibrium between security and scalability, where algorithmic cohesion replaces centralized authority [79].

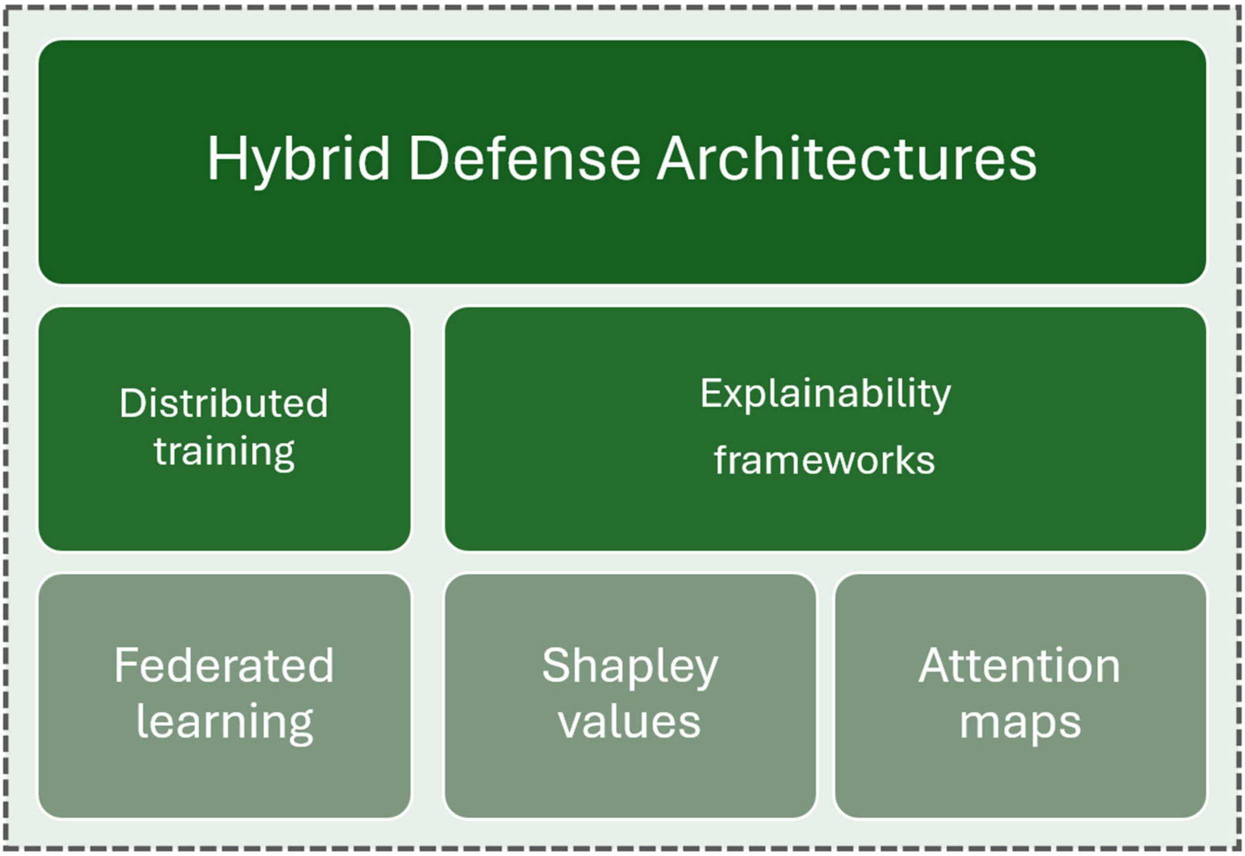

Figure 7. Taxonomy of Hybrid Defense Architectures in Cybersecurity. This diagram highlights advanced architectures that blend algorithmic performance with system-level robustness. The distributed training branch includes federated learning, which enables collaborative model training across decentralized data sources without compromising privacy. The explainability frameworks branch comprises Shapley values and attention maps, which help interpret and visualize model decisions, that are essential for auditing, debugging, and building trust in AI-driven defenses.

Furthermore, Explainable Artificial Intelligence (XAI) injects transparency into this algorithmic amalgam that brings the knowledge gap between human analysts and opaque models. Explainability frameworks use Shapley values, attention maps, and symbolic surrogates to trace decision boundaries and restore trust in high-stakes defensive automation [80]. In this hybrid landscape, interpretability is not a concession to ethics but a mechanism of robustness, since adversarial exploitation often thrives on hidden logic [81]. The result is an emergent defense topology where symbolic precision meets statistical fluidity, namely an algorithmic symbiosis that transforms cyber defense from reactive computation into adaptive cognition [82].

The cross-domain approach of this review complements several recent task-specific surveys. For example, Tagarev and Sharkov [83] address system-level security in distributed contexts, Chunawala et al. [84] focus on graph-based anomaly detection, and Ali et al. [85]

highlight quantum-era cryptographic threats. Other reviews have concentrated on cyber threat intelligence in IoT systems [86] and anomaly detection in cyber-physical infrastructures [87]. In contrast, this work unifies diverse algorithmic paradigms through an integrative lens, that offers a consolidated view of defense strategies under adversarial conditions [88].

8. Conclusions

The evolution of cyber defense reflects the convergence of mathematical precision, algorithmic adaptation, and systemic cognition. Cryptographic algorithms define the immutable core of confidentiality, while detection and learning systems embody the mutable edge of behavioral intelligence. The integration of classical computation with adaptive learning architectures transforms security from a reactive discipline into an anticipatory science capable of evolving alongside its adversaries. Graph-theoretic reasoning, probabilistic inference, and reinforcement learning together establish a new algorithmic dialect, one in which defense is expressed as computation over uncertainty. Future cyber defense will depend not merely on faster algorithms but on architectures that learn, reason, and explain their decisions autonomously. The synthesis presented here repositions algorithms as the true genomic code of cyber defense: entities that evolve, hybridize, and replicate strategic intelligence across digital ecosystems.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Advanced Encryption Standard
ECC	Elliptic Curve Cryptography
RSA	Rivest-Shamir-Adleman
ECDH	Elliptic Curve Diffie-Hellman
HMM	Hidden Markov Model
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
SIEM	Security Information and Event Management
DQN	Deep Q-Network
AIS	Artificial Immune System
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
CFG	Control Flow Graph
DFG	Data Flow Graph
SAT	Boolean Satisfiability Problem
SMT	Satisfiability Modulo Theories
XAI	Explainable Artificial Intelligence
PQC	Post-Quantum Cryptography
SHA	Secure Hash Algorithm
DHT	Distributed Hash Table
HIDS	Host-Based Intrusion Detection System
NIDS	Network-Based Intrusion Detection System

Appendix A

Taxonomic Rationale

The taxonomies presented in this review are original and constructed by the author. Their development was guided by three principal dimensions: (i) the functional role of each algorithm within the cyber defense ecosystem (e.g., encryption, intrusion detection, behavioral inference); (ii) the mathematical paradigm underlying the algorithm (e.g., deterministic, probabilistic, adaptive); and (iii) the capacity of each algorithm to sustain performance when exposed to adversarial inputs, as evaluated by factors such as error tolerance, transparency of decision-making, and ability to adjust to non-stationary data distributions. Therefore, this framework may allow for a structured and cross-cutting classification that aligns foundational principles with operational relevance in cybersecurity.

References

1. Bishop, M. *Security, Computer Security: Art and Science*, 2nd ed.; Addison-Wesley: Boston, MA, USA, 2018.
2. Paul, A.G. *Antivirus Engines: From Methods to Innovations and Applications*; Elsevier Syngress: Burlington, MA, USA, 2024; pp. 1–656.
3. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 8th ed.; Pearson: London, UK, 2023.
4. Axelsson, S. The base-rate fallacy and its implications for the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur.* **1999**, *3*, 186–205. [CrossRef]
5. Bishop, C. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
6. Song, Y. Application of Deep Learning in Malware Detection: A. Review. *J. Big Data* **2025**, *12*, 57. [CrossRef]
7. Rivest, R.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
8. Ozkan-Okay, M.; Akin, E.; Aslan, Ö.; Kosunalp, S.; Iliev, T.; Stoyanov, I.; Beloev, I. A Comprehensive Survey: Evaluating the Efficiency of Artificial Intelligence and Machine Learning Techniques on Cyber Security Solutions. *IEEE Access* **2024**, *12*, 12252–12278. [CrossRef]
9. Boneh, D.; Shoup, V. *A Graduate Course in Applied Cryptography*; Stanford University: Stanford, CA, USA, 2020.
10. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–26 May 2017; pp. 39–57.
11. Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 1996.
12. Daemen, J.; Rijmen, V. *The Design of Rijndael: AES—The Advanced Encryption Standard*; Springer: Berlin/Heidelberg, Germany, 2002.
13. *FIPS PUB 197*; Advanced Encryption Standard (AES). National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2001.
14. Bernstein, D.J. *ChaCha, a Variant of Salsa20 Workshop Record of SASC 2008*; The State of the Art of Stream Ciphers; The University of Illinois at Chicago: Chicago, IL, USA, 2008.
15. Williams, H.C. A Modification of the RSA Public-Key Encryption Procedure. *IEEE Trans. Inf. Theory* **1980**, *26*, 726–729. [CrossRef]
16. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
17. Miller, V.S. *Use of Elliptic Curves in Cryptography. Advances in Cryptology—CRYPTO '85 Proceedings*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 417–426.
18. Bertoni, G.; Daemen, J.; Peeters, M.; Van Assche, G. Keccak. In *Advances in Cryptology—EUROCRYPT 2013*; Johansson, T., Nguyen, P.Q., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7881, pp. 313–314.
19. O'Connor, J.; Aumasson, J.P.; Neves, S.; Wilcox-O'Hearn, Z. BLAKE3: One Function, Fast Everywhere Cryptology ePrint Archive, Report 2020/014. 2020. Available online: https://autoblog.suumitsu.eu/autoblogs/sebsauvagenetlinks_f5730e92d2c28615f2f44758a6d249f605e9c0ae/media/f32098b6.blake3.pdf (accessed on 20 October 2025).
20. Biryukov, A.; Dinu, D.; Khovratovich, D. Argon2: The memory-hard function for password hashing and other applications. In Proceedings of the IEEE EuroS&P, Saarbruecken, Germany, 21–24 March 2016; pp. 292–302.
21. Leo, D.; Tancrede, L.; Vadim, L.; Peter, S.; Gregor, S.; Damien, S. CRYSTALS—Dilithium: Digital Signatures from Module Lattices. IACR ePrint Archive, 2017/633. 2017. Available online: https://eprint.iacr.org/2017/633 (accessed on 12 January 2025).
22. Bernstein, D.J.; Dobraunig, C.; Eichlseder, M.; Fluhrer, S.; Gazdag, S.-L.; Hülsing, A.; Kampanakis, P.; Kölbl, S.; Lange, T.; Lauridsen, M.M.; et al. *SPHINCS+: Submission to the NIST Post-Quantum Cryptography Project*; NIST PQC Round 3 Submission; Eindhoven University of Technology: Eindhoven, The Netherlands, 2020.
23. Peikert, C. A decade of lattice cryptography. Found. Trends Theor. *Comput. Sci.* **2016**, *10*, 283–424.

24. Farhan, M.; Waheed Ud Din, H.; Ullah, S.; Hussain, M.S.; Khan, M.A.; Mazhar, T.; Jaghdam, I.H. Network-based intrusion detection using deep learning technique. *Sci. Rep.* **2025**, *15*, 25550. [CrossRef]
25. Aho, A.V.; Corasick, M.J. Efficient string matching: An aid to bibliographic search. *Commun. ACM* **1975**, *18*, 333–340. [CrossRef]
26. Boyer, R.S.; Moore, J.S. A fast string searching algorithm. *Commun. ACM* **1977**, *20*, 762–772. [CrossRef]
27. Panda, M.; Patra, M.R. Network intrusion detection using Naive Bayes. *Int. J. Comput. Sci. Netw. Secur* **2007**, *7*, 258–263.
28. Gagniuc, P.A. *Markov Chains: From Theory to Implementation and Experimentation*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2017.
29. Kim, K.L.; Kuperman, B.A. Using Hidden Markov Models for intrusion detection. In Proceedings of the IEEE Systems Man and Cybernetics Society, Taipei, Taiwan, 8–11 October 2006; pp. 1756–1761.
30. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
31. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
32. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
33. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
34. An, J.; Cho, S. *Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability*; SNU Data Mining Center Technology Report; Seoul National University: Seoul, Republic of Korea, 2015.
35. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the ICLR, Banff, AB, Canada, 14–16 April 2014.
36. Altaha, S.J.; Aljughaiman, A.; Gul, S. A Survey on Android Malware Detection Techniques Using Supervised Machine Learning. *IEEE Access* **2024**, *12*, 173168–173191. [CrossRef]
37. Kruegel, C.; Kirda, E.; Mutz, D.; Robertson, W.; Vigna, G. Polymorphic worm detection using structural information of executables. In Proceedings of the RAID, Seattle, WA, USA, 7–9 September 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 207–226.
38. Nielson, F.; Nielson, H.R.; Hankin, C. *Principles of Program Analysis*; Springer: Berlin/Heidelberg, Germany, 2005.
39. Vasudevan, A. Re-inforced Stealth Breakpoints. In Proceedings of the 2009 Fourth International Conference on Risks and Security of Internet and Systems (CRiSIS 2009), Toulouse, France, 19–22 October 2009; pp. 59–66.
40. Lyda, R.; Hamrock, J. Using entropy analysis to find encrypted and packed malware. *IEEE Secur. Privacy* **2007**, *5*, 40–45. [CrossRef]
41. Christodorescu, M.; Jha, S. Static analysis of executables to detect malicious patterns. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 4–8 August 2003; pp. 169–186.
42. Aboaoja, F.A.; Zainal, A.; Ghaleb, F.A.; Al-Rimy, B.A.S.; Eisa, T.A.E.; Elnour, A.A.H. Malware Detection Issues, Challenges, and Future Directions: A Survey. *Appl. Sci.* **2022**, *12*, 8482. [CrossRef]
43. Berrios, S.; Leiva, D.; Olivares, B.; Allende-Cid, H.; Hermosilla, P. Systematic Review: Malware Detection and Classification in Cybersecurity. *Appl. Sci.* **2025**, *15*, 7747. [CrossRef]
44. Moura, L.; Bjørner, N. Z3: An Efficient SMT Solver. In Proceedings of the TACAS, Budapest, Hungary, 29 March–6 April 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 337–340.
45. Eén, N.; Sörensson, N. An Extensible SAT-Solver. In Proceedings of the SAT, Santa Margherita Ligure, Italy, 5–8 May 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 502–518.
46. Salwan, J.; Bardin, S.; Potet, M.L. Symbolic Deobfuscation: From Virtualized Code Back to the Original. In *DIMVA 2018: Detection of Intrusions and Malware, and Vulnerability Assessment*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 372–392.
47. Dullien, T. Graph-Based Comparison of Executable Objects (Reloaded). In Proceedings of the Symposium Sur la Sécurité des Technologies de L'information et des Communications, Rennes, France, 30 May–1 June 2007.
48. Newman, M. *Networks: An Introduction*; Oxford University Press: Oxford, UK, 2010.
49. Barbarossa, S.; Scutari, G. Distributed signal processing in self-organizing sensor networks. *IEEE J. Sel. Areas Commun.* **2007**, *24*, 775–786.
50. Zhong, X.; Sun, Y.; Li, Q. A Survey on Graph Neural Networks for Intrusion Detection Systems: Methods, Trends and Challenges. *Comput. Secur.* **2024**, *125*, 103152. [CrossRef]
51. Floyd, R.W. Algorithm 97: Shortest path. *Commun. ACM* **1962**, *5*, 345. [CrossRef]
52. Shapiro, J.; Sandhu, R.S. A Weighted Shortest-Path Approach to Network Security Metrics. In Proceedings of the IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; pp. 1642–1647.
53. Ford, L.R.; Fulkerson, D.R. Maximal flow through a network. *Canad. J. Math.* **1956**, *8*, 399–404. [CrossRef]
54. Edmonds, J.; Karp, R.M. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* **1972**, *19*, 248–264. [CrossRef]
55. Sperotto, A.; Schaffrath, G.; Sadre, R.; Morariu, C.; Pras, A.; Stiller, B. Flow-Based Intrusion Detection: From Theory to Practice. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection (RAID), Dublin, Ireland, 23–27 May 2011; pp. 1–17.
56. Yu, S.Z.; Zhang, Y.X.; Wang, C. Botnet detection using improved PageRank algorithm. *J. Netw. Comput. Appl.* **2013**, *36*, 1441–1450.
57. Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef]

58. Zhang, J.; Zhou, S. Detecting coordinated attacks in large networks using community detection. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1533–1547.
59. Dainotti, A.; Pescape, A.; Claffy, K.C. Issues and future directions in traffic classification. *IEEE Netw.* **2012**, *26*, 35–40. [CrossRef]
60. Nguyen, H.; Altman, E.; Nain, P. Reinforcement learning in network security: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1649–1697.
61. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
62. Watkins, C.J.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
63. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Hassabis, D. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
64. Zhu, H.; Wang, W.; Kim, H. Adaptive cyber defense using deep reinforcement learning. In Proceedings of the IEEE Security Privacy Workshops (SPW), Genoa, Italy, 21 May 2020; pp. 8–14.
65. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Boston, MA, USA, 1989.
66. De Castro, L.N.; Von Zuben, F.J. *Artificial Immune Systems: Part I—Basic Theory and Applications*; Technical Report TR-DCA 01/99; State University: Campinas, Brazil, 1999.
67. Forrest, S.; Perelson, A.S.; Allen, L.; Cherukuri, R. Self-Nonself Discrimination in a Computer. In Proceedings of the IEEE Symposium Security Privacy, Oakland, CA, USA, 16–18 May 1994; pp. 202–212.
68. Dorigo, M.; Stützle, T. *Ant Colony Optimization*; MIT Press: Cambridge, MA, USA, 2004.
69. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
70. Paranjape, S.; Chandane, P.; Chatur, P. A hybrid swarm intelligence approach for network intrusion detection. *Expert Syst. Appl.* **2022**, *206*, 118011.
71. Bridges, S.M.; Vaughn, R.B. Fuzzy data mining and genetic algorithms applied to intrusion detection. In Proceedings of the 12th Annual Canadian Information Technology Security Symposium. Workshop, Baltimore, MD, USA, 16–19 October 2000; pp. 109–122.
72. Cohen, D.; Te'eni, D.; Yahav, I.; Zagalsky, A.; Schwartz, D.; Silverman, G.; Makowski, J. Human–AI Enhancement of Cyber Threat Intelligence. *Int. J. Inf. Secur.* **2025**, *24*, 99. [CrossRef]
73. Lilhore, U.K.; Simaiya, S.; Alroobaea, R.; Baqasah, A.M.; Alsafyani, M.; Alhazmi, A.; Khan, M.M. SmartTrust: A hybrid deep learning framework for real-time threat detection in cloud environments using Zero-Trust Architecture. *J. Cloud Comput.* **2025**, *14*, 35. [CrossRef]
74. Kim, S.; Shin, J.; Kim, H. Hybrid intrusion detection using deep neural networks and Markov models. *IEEE Access* **2021**, *9*, 73182–73193.
75. Ye, Y.; Li, T.; Adjeroh, D.; Iyengar, S.S. A survey on malware detection using data mining and machine learning. *ACM Comput. Surv.* **2017**, *50*, 41.
76. Han, Z.; Niyato, D.; Saad, W.; Başar, T. *Game Theory for Next Generation Wireless and Communication Networks*; Cambridge University Press: Cambridge, UK, 2019.
77. Konečný, J.; McMahan, H.B.; Ramage, D.; Richtárik, P. Federated Optimization: Distributed Machine Learning for on-Device Intelligence. *arXiv* **2016**, arXiv:1610.02527. [CrossRef]
78. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 12. [CrossRef]
79. Mo, K.; Tang, W.; Li, J.; Yuan, X. Cloud-based distributed intrusion detection using ensemble learning. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 233–247.
80. Guo, W.; Mu, D.; Xu, J.; Su, P.; Wang, G.; Xing, X. LEMNA: Explaining Deep Learning based Security Applications. In Proceedings of the 2018 ACM SIGSAC Conference on Computer & Communications Security (CCS '18), Toronto, ON, Canada, 15 October 2018; pp. 364–379.
81. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.* **2018**, *84*, 317–331. [CrossRef]
82. Buczak, A.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
83. Tagarev, T.; Sharkov, G. Computationally Intensive Functions in Designing and Operating Distributed Cyber Secure and Resilient Systems. In Proceedings of the 20th International Conference on Computer Systems and Technologies (CompSysTech), New York, NY, USA, 21–22 June 2019; pp. 8–18.
84. Chunawala, H.; Kumbhar, S.; Pandey, A.; Rajput, B.J.; Sahu, G.; Guru, A. A Survey of Anomaly Detection in Graphs: Algorithms and Applications. In *Graph Mining*; Bhattacharya, R., Rathore, Y.K., Tran, T.A., Swarnkar, S.K., Eds.; Springer: Cham, Switzerland, 2025.

85. Ali, S.; Wadho, S.A.; Talpur, K.R.; Talpur, B.A.; Alshudukhi, K.S.; Humayun, M.; Shah, A. Next-generation quantum security: The impact of quantum computing on cybersecurity-Threats, mitigations, and solutions. *Comput. Electr. Eng.* **2025**, *128*, 110649. [CrossRef]
86. Lima, M.; Viana, C.; Santos, W.R.; Neves, F.; Campos, J.R.; Aires, F. Toward using cyber threat intelligence with machine and deep learning for IoT security: A comprehensive study. *J. Supercomput.* **2025**, *81*, 1404. [CrossRef]
87. Moriano, P.; Hespeler, S.C.; Li, M.; Mahbub, M. Adaptive anomaly detection for identifying attacks in cyber-physical systems: A systematic literature review. *Artif. Intell. Rev.* **2025**, *58*, 283. [CrossRef]
88. Bogdan, I.C.; Simion, E. Cybersecurity Assessment and Certification of Critical Infrastructures. *Univ. Politeh. Buchar. Sci. Bull. Ser. C* **2024**, *86*, 4.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.