A Bio-Inspired Behavior-Based Hybrid Framework for Ransomware Detection

Mohammed A. F. Salah[1], Mohd Fadzli Marhusin[2], Rossilawati Sulaiman[3]
Faculty of Information Science and Technology-Center for Cyber Security, Universiti Kebangsaan Malaysia, Bangi Selangor, Malaysia[1, 3]
Faculty of Sciences and Technology-Cyber Security and Systems (CSS) Research Unit, Universiti Sains Islam Malaysia, 71800 Nilai, Negeri Sembilan, Malaysia[2]

***Abstract*—Ransomware remains a critical and evolving cybersecurity threat, increasingly rendering traditional signature-based detection techniques ineffective. While modern machine learning models achieve high detection accuracy, they often operate as opaque "black boxes", introducing a significant explainability gap that undermines analyst trust. In addition, behavior-based anomaly detection systems frequently suffer from high false-positive rates, limiting their operational viability. To address these challenges, this study adopts a Design Science Research Methodology to develop a novel, interpretable, multi-stage ransomware detection framework. The proposed architecture integrates three complementary components: a bio-inspired Negative Selection Algorithm from Artificial Immune Systems to filter benign behavioral patterns, a first-order Markov chain model to capture probabilistic deviations in execution sequences, and a Random Forest ensemble classifier to synthesize these signals for final decision-making. The framework is evaluated using a dual-pipeline experimental design on real-world ransomware and benign software samples, enabling controlled comparison between probabilistic and pattern-based behavioral modeling. Experimental results demonstrate that the proposed approach achieves high detection performance while maintaining a low false-positive rate and providing interpretable behavioral evidence. Overall, the framework offers a principled balance between detection effectiveness and interpretability, addressing key limitations of existing ransomware detection systems.**

Keywords*—Ransomware; Artificial Immune Systems (AIS); anomaly detection; Negative Selection Algorithm; Markov chain; Random Forest; hybrid framework

I. Introduction

Computers and the Internet have revolutionized many aspects of daily life, offering unprecedented opportunities for communication, information exchange, and economic growth [1]. This digital transformation, accelerated by rapid advancements in information technology, has significantly improved global living standards [2]. However, the rise of the digital age has also presented new challenges in the form of cyber threats [3], [4].

Ransomware has rapidly transformed into a significant global cybersecurity challenge by employing advanced cryptographic techniques, polymorphic code mutations, and anti-analysis mechanisms to avoid detection [5], [6], [7]. The increasing sophistication of these attacks renders the traditional signature-based detection systems ineffective. These static pattern-matching approaches are fundamentally unable to combat the dynamic and evasive nature of 0-day and polymorphic ransomware variants [8], [9], [10].

High-profile incidents, such as the 2021 Colonial Pipeline attack, demonstrate the devastating impact of ransomware on critical infrastructures. This incident highlighted severe vulnerabilities and led to a notable fuel shortage along the US East Coast [11]. This threat is further exacerbated by the "Ransomware-as-a-Service" (RaaS) model, which enables cybercriminals with minimal technical expertise to launch large-scale, sophisticated attacks [12].

In response to the failure of static methods, the research community has increasingly focused on dynamic and behavior-based malware detection using machine learning (ML). behavior-based approaches inspired by biological systems, particularly the Human Immune System, have demonstrated promise in detecting previously unseen malware variants through adaptive and anomaly-driven mechanisms [13]. However, despite their effectiveness, existing immune-inspired and hybrid detection architectures often remain limited in their ability to robustly model complex runtime behavior and to adapt to the evolving tactics of modern ransomware [14]. This motivates the need for a more formalized and empirically validated behavior-centric detection framework.

Our systematic review of the domain from 2015 to 2025 reveals three critical gaps that impede the deployment of robust real-world defenses:

- The Explainability Gap (G1): Many high- performing models, particularly deep learning architectures, operate as "black boxes" [15]. Security analysts cannot trust or act on alerts that they do not understand, making interpretability a critical, unsolved requirement [16].
- The False Positive Gap (G2): Behavior-based anomaly detection is notoriously "noisy". Legitimate software, such as file compressors and backup tools, often performs intensive I/O operations that mimic ransomware [17]. This leads to a high false positive rate (FPR), and the resulting alert fatigue renders many systems unusable in operational contexts [18].
- The AIS-ML Gap (G3): While Artificial Immune Systems (AIS), particularly the Negative Selection Algorithm (NSA), are theoretically ideal for the "self" vs. "non-self" discrimination problem, their application in ransomware detection remains limited and non-

standardized [19]. The value of AIS-ML hybrids is a significant and under-explored research area.

To address these three gaps, this study proposes a novel, interpretable, AIS-hybrid framework for ransomware detection. Following a rigorous Design Science Research Methodology (DSRM) [20], we present the complete design artefact of a system specifically engineered to solve these problems.

Our contributions are:

- Novel Hybrid Detection Architecture: We propose a bio-inspired, behavior-based ransomware detection framework that integrates an Artificial Immune System (Negative Selection Algorithm), probabilistic Markov chain behavioral profiling, and Random Forest classification in a unified, multi-stage architecture. Unlike prior studies that employ these techniques in isolation or simple pairwise combinations, this work formalizes their complementary integration into a single explainable detection pipeline.
- AIS-Based Behavioral Noise Reduction: We introduce an NSA-driven behavioral filtering mechanism that removes benign "self" API-call patterns prior to classification, substantially reducing behavioral noise and mitigating false positives commonly observed in anomaly-based ransomware detectors.
- Probabilistic Behavioral Explainability: We incorporate a first-order Markov chain model trained on benign execution traces to quantify the likelihood of observed API-call transitions. This probabilistic profiling enables interpretable detection decisions by highlighting statistically rare behavioral sequences rather than opaque feature activations.
- Controlled Dual-Pipeline Evaluation: We design and evaluate two detection pipelines under identical experimental conditions, NSA–Markov–RF and NSA–n-gram–RF, allowing a controlled assessment of the contribution of probabilistic behavioral modeling to detection performance and interpretability.

The remainder of this study is organized as follows: Section II reviews related work and identifies key research gaps. Section III presents the theoretical framework underlying the proposed hybrid detection architecture. Section IV describes the proposed methodology, including evaluation metrics and comparative pipeline design. Section V reports the experimental evaluation, covering dataset construction, experimental setup, detection results, and analysis of the Markov behavioral modeling contribution. Section VI discusses the theoretical and practical implications of the findings, including threat modeling, computational feasibility, and future research directions. Finally, Section VII concludes the study.

II. Literature Review

A. Ransomware Detection Techniques

Ransomware detection methods rely heavily on signature-based and heuristic techniques [21], [22]. Signature matching fails against obfuscated or polymorphic variants [23], [24]. Heuristic approaches improve this by looking for suspect behaviors, such as mass file encryption [25], but they suffer from high false positive rates (FPR), as benign software, that is, backup tools that can exhibit similar behaviors [26], [27].

To combat these novel threats, researchers have turned to machine learning for behavioral data [5]. Dynamic analysis frameworks such as Cuckoo/CAPE sandboxes were used to extract runtime API call sequences as features for ML models [28], [29]. However, these ML-driven methods introduce their own critical challenges, namely: 1) Limited Explainability (G1), as many models are "black boxes", and 2) Persistent False Positives (G2), as benign apps still mimic ransomware behavior.

B. Machine Learning and Ensembles

A range of ML algorithms has been successfully applied to ransomware detection. Tree-based Decision methods (J48, C5.0) and ensemble methods (Random Forest, gradient boosting) have shown high accuracy in classifying ransomware vs. good-ware binaries [30], [31]. Moreover, [32] used a Random Forest classifier on extracted file features and achieved ~97.7% accuracy in distinguishing ransomware, although with a somewhat elevated false-positive rate in practical scenarioslink.springer.com. Random Forest (RF), in particular, has become popular owing to its strong performance and resistance to overfitting. It combines multiple decision trees to vote on the class, which tends to smooth out noise and outlier effects.

Khammas's results illustrate both the promise and pitfall of pure ML: high accuracy is attainable, but the cost of false alarms can be significant if the model latches onto features that overlap with legitimate softwarelink.springer.com. Recent studies have emphasized explainable machine learning. For instance, techniques like SHAP (Shapley Additive Explanations (SHAPs) have been employed to interpret feature importance in ransomware classifiersarxiv.orgarxiv.org, and uncertainty estimation has been added to random forests to identify low confidence. These trends highlight the growing recognition that detection models must be not only accurate but also transparent and reliable for analysts. This limitation of pure ML-only models was exemplified in the work of [21]. In their initial research, they proposed a lightweight 10-feature Random Forest model. However, to improve the performance in their follow-up work, they were forced to hybridize this model with a static, signature-based hash filter [33]. This progression demonstrates the practical limitations of the pure ML approach (G2) and highlights the need for a more dynamic, adaptive hybrid. While [33] resorted to a static filter, our framework proposes a novel hybrid with a true adaptive technique (AIS) to solve this exact problem, directly addressing the G3 gap.

C. Markov Chain Models in Malware Behavior

Markov models provide a stochastic way to model sequences of events, which is highly relevant for ransomware behavior (often characterized by sequences of file accesses or API calls). Researchers have applied both Markov chains [34] and Hidden Markov Models (HMMs) [35] for malware and ransomware analyses. An early example was the use of Markov chains to model system call sequences for anomaly detection in host intrusion detection systems. In ransomware-specific research, [36] proposed a two-stage detection approach, first modelling API call sequences with Markov chains, and then applying a

Random Forest (RF) classifier. Their system trained separate models for benign and ransomware behaviors to capture the malicious transition probabilities. Despite achieving 97.3% accuracy, their hybrid model still suffered from a non-negligible 4.8% false-positive rate (FPR). This result confirms that while Markov models can encapsulate sequential behavior, a high FPR remains a key drawback. Crucially, the authors suggested that "additional anomaly detection" could reduce these false alarms [37].

This insight provides a direct motivation for our framework, which integrates an AIS-based anomaly detector, the NSA, as the precise component required to filter out benign patterns and solve this identified gap. Although HMM-based approaches can classify ransomware families, they often require substantial training data and are computationally heavy. This is precisely why our framework leverages a first-order Markov Chain, which provides the significant advantage of being lightweight and interpretable. A simple transition probability matrix allows a security analyst to examine which specific API call transitions are deemed 'rare' or suspicious, directly addressing the G1 (Explainability) gap.

D. Artificial Immune Systems (AIS) and Negative Selection

Artificial Immune System (AIS) algorithms are inspired by the human immune system's ability to distinguish "self" (normal body cells) from "non-self" (foreign pathogens). In AIS, the negative-selection algorithm (NSA) is one of the most prominent techniques for anomaly detection. Originally introduced by [38], [39], the NSA generates a set of detectors that do not match any known "self" patterns. During detection, any pattern that failed to match the self-set was flagged as an anomaly. NSA has been applied to cybersecurity to detect novel threats, with some work combining it with positive selection to improve the detection of unknown malware [40].

In the context of ransomware, AIS has been shown to be effective. For example, [41] developed a V-detector AIS model for ransomware that learns to divide sequence space. Moreover, [42] proposed an improved double-layer NSA that achieved a higher coverage of non-self-space and improved detection rates. The success of these studies demonstrated that bio-inspired methods can enhance ransomware anomaly detection. However, AIS methods alone typically provide only binary anomaly judgments. For example, the NSA can identify a pattern as "suspicious" but cannot classify why or if it is specifically ransomware. This limitation is precisely why the hybrid approach is very powerful. By combining AIS with machine learning, AIS can act as an upstream filter or feature generator for a more robust ML classifier.

To explicitly position the proposed framework with respect to existing ransomware detection approaches, Table I presents a conceptual comparison highlighting key architectural and design differences.

TABLE I. CONCEPTUAL COMPARISON OF THE PROPOSED FRAMEWORK WITH REPRESENTATIVE RANSOMWARE DETECTION APPROACHES

Study	Behavior	AIS/ NSA	Markov	ML	Explainable	FB
[29]	Partial	✗	✗	RF	✗	Static hash filtering
[32]	✓	✗	✓	RF	Limited	✗
[37]	✓	✓	✗	✗	Limited	Partial
[38]	✓	✓	✗	✗	✗	Partial
This work	✓	✓	✓	✓	✓	✓

III. THEORETICAL FRAMEWORK

A. Overview

The proposed framework builds upon earlier immune-inspired malware detection architectures that employed a multi-stage design and Negative Selection–based anomaly filtering [13]. While prior work established the feasibility of immune-system-inspired detection, it did not explicitly investigate fine-grained sequential behavioural modelling or adaptive feature representations for ransomware detection. This study extends those ideas by introducing a hybrid framework that integrates NSA filtering, sequential modelling, and machine-learning-based classification, supported by rigorous experimental evaluation.

The proposed ransomware detection framework follows a multilayered architecture inspired by immune system principles and probabilistic modelling.

At a high level, it can be conceptualized as a dual-stage pipeline: an anomaly detection stage that flags potentially malicious behavior and a classification stage that confirms and labels the ransomware. These stages correspond to the Negative Selection Algorithm and the Random Forest classifier, respectively, with a Markov Chain model bridging them by providing probabilistic features of the behavior. Fig. 1 (conceptual diagram) illustrates how raw execution traces flow through the layers of the system from data acquisition to the final decision. The architecture is modular, meaning that each component has a distinct role and a well-defined input-output interface, which aids both in conceptual clarity and in actual implementation.

B. Data and Feature Representation

We assume as input the dynamic behavior traces of programs, specifically focusing on the API call sequences observed during program execution. This choice is guided by extensive evidence that API call patterns are highly indicative of ransomware behavior. Ransomware typically invokes cryptographic APIs, file system calls, etc. [43], [44]. Let an execution trace be represented as a sequence of API call events: $S = [s_1, s_1, \ldots, s_n]$, s_i is an API function name or a token representing a specific action. In our framework, we derive two parallel representations from this sequence: 1) a set of binary features or anomaly markers from the NSA indicating whether certain anomalous patterns were detected and 2) a set of

probabilistic features from the Markov chain model quantifying how likely each observed transition is under a normal profile. These features were then concatenated to form a composite feature vector for classification.

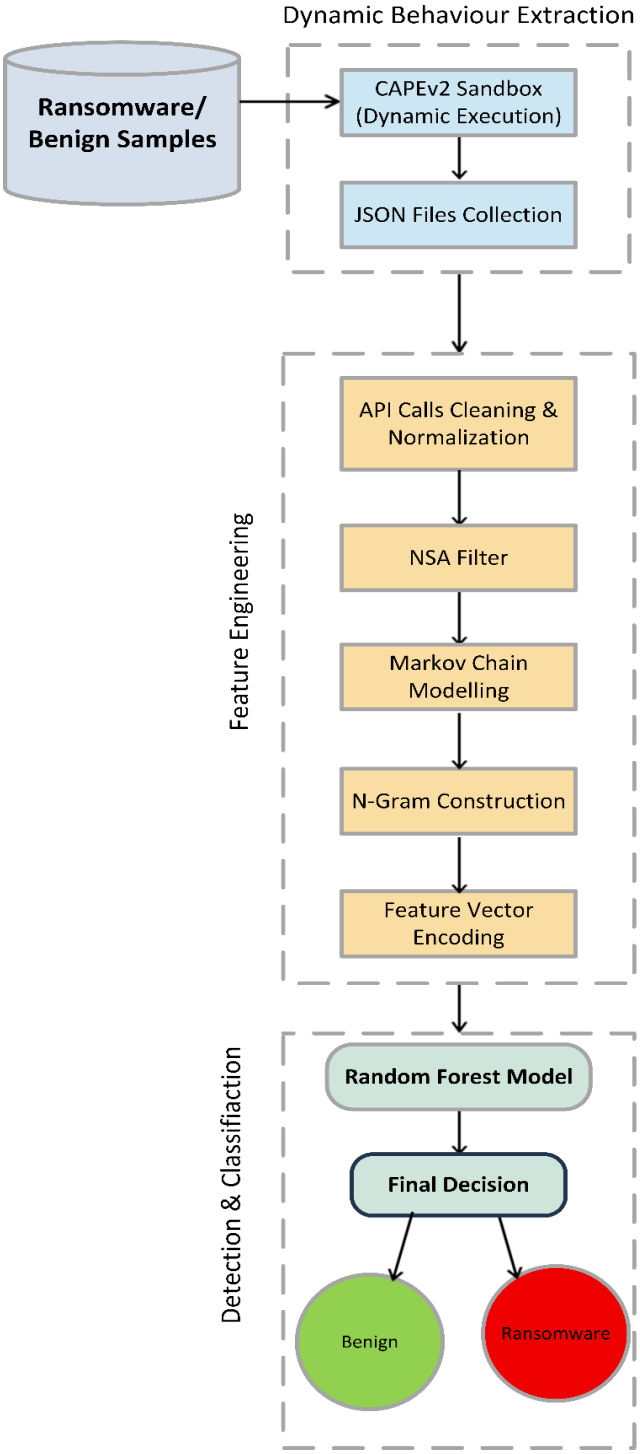

Fig. 1. The proposed NSA-Markov-RF theoretical framework (Pipeline 1).

C. Negative Selection Algorithm (NSA) Component

The NSA is responsible for learning the profile of normal (benign) API call sequences and generating detectors for any sequence patterns that deviate from this norm. Formally, we define a self-set $\mathcal{S}$ consisting of abstractions of benign API sequences. Each element of $\mathcal{S}$ could be, for example, an n-gram of API calls or some fixed-length substring of normal sequences, encoded as a string or binary vector. The NSA then creates a collection of detectors $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ such that none of the detectors matches any element of $\mathcal{S}$ within a certain matching rule or threshold. A common approach is to use r-contiguous bit matching for the binary-encoded strings [42]. In our context, we might encode API calls or their hashed values in binary format and generate detectors that do not occur in any benign trace up to a certain Hamming distance. During detection, for a given sequence S, the NSA scans for any detector $d \in \mathcal{D}$ that is a substring of S or matches some features of S. If any such detector is found, it signals the presence of a non-self-pattern, that is, a potential anomaly.

Mathematically, let $f: \text{Sequence} \rightarrow \{0,1\}^m$ be a function that maps an input sequence to a binary vector of detector activations, where the k-th component $f_k(S) = 1$ if and only if detector d_k matches the sequence S. The output of the NSA stage can then be considered an anomaly feature vector $A = f(S)$. If all components of A are 0, the sequence was entirely self-like (no anomalies detected); if one or more components are 1, those indicate specific detectors (specific patterns) that were triggered. We emphasize that NSA in our framework serves as a feature generator and filter, rather than a final decision maker: the presence of an anomaly pattern raises suspicion but does not by itself convict the program as ransomware. This helps reduce false positives, as benign outliers might occasionally trigger detectors but will be further scrutinized in later stages.

D. Markov Chain Model

In parallel with the NSA analysis, the framework uses a First-Order Markov Chain to model the likelihood of the observed sequence of API calls. A first-order Markov chain assumes that the probability of each API call depends only on the immediately preceding call. We derive a Transition Probability Matrix (TPM) P of size $N \times N$, where N is the number of unique API calls (states). Each entry $P_{ij} = \Pr(s_{t+1} = j \mid s_t = i)$ represents the probability of transitioning from API call i to API call j in a benign execution trace. We estimate these probabilities from a training set of benign sequences by frequency counting (i.e., maximum likelihood estimation for the Markov chain). Let $C(i,j)$ denote the count of transitions from state i to j in all benign traces. Then the estimated transition probability is:

$$P_{ij} = \frac{C(i,j)}{\sum_k C(i,k)} \quad (1)$$

for each state i that appears in the training data (the denominator is the total count of all transitions out of state i). By construction, for each state i, the probabilities sum to 1 ($\sum_j P_{ij} = 1$), making each row of P a distribution over next state. Given the matrix P, the sequence $S = [s_1, \ldots, s_n]$ from a program run can be assigned a likelihood score under the benign-behavior model:

$$\Pr(S \mid \text{benign-model}) = \prod_{t=1}^{n-1} P_{s_t s_{t+1}}, \quad (2)$$

i.e., the product of probabilities for each consecutive transition in the sequence. If this probability is very low or equivalently the log-likelihood sum is very low, it indicates that the sequence contains transitions that were rarely or never observed in benign runs – a strong sign of anomalous (possibly malicious behavior. In practice, rather than using a single product of probabilities, which can underflow and also over-emphasize long sequences, we extract features from the Markov model, such as minimum transition probability observed in the sequence, average transition probability, and number of transitions below a certain low-probability threshold. Each of these features helps quantify how "normal" or "abnormal" the control flow of the program is, according to the benign profile. For example, a ransomware might introduce an API call sequence that jumps between unrelated system functions in a way a normal program never would, yielding a very low P_{ij} for those transitions. By including such metrics, we capture sequential anomalies different from those the NSA detectors find, where NSA might catch specific known-bad patterns, whereas Markov probabilities catch statistically rare patterns.

It is worth noting that one could also build a Markov chain for ransomware behavior, specifically as [36] did, training separate models for ransomware and benign. In our theoretical design, we focus on modeling benign behavior (self) through the Markov chain, analogous to how NSA models self; this keeps the approach biased towards anomaly detection. However, extending the framework to include a second Markov model for ransomware is feasible if sufficient ransomware data is available; it would essentially provide additional features or a parallel likelihood ratio between "malicious" and "benign" models.

E. Random Forest Classification

Random Forest Classification: The final stage of the pipeline is a Random Forest classifier that ingests the combined feature vector and outputs a classification: ransomware or benign. The feature vector X for a given program execution includes features from both NSA and the Markov model and potentially other metadata or features derived from dynamic analysis. For example, X might include: the count of NSA detectors triggered and which specific ones, though high-dimensional, could be represented via one-hot encoding or a count of anomalies, the minimum/average transition probability from the Markov analysis, the length of the sequence, etc. Because NSA and Markov modeling are the core novel features, the dimensionality of this feature space is relatively moderate and interpretable: NSA contributes m binary features, if we have m detectors, and Markov contributes a fixed number of statistical features. This structured feature space is well-suited to a Random Forest classifier, which can handle mixed binary/continuous inputs and is robust to irrelevant features.

Random Forest (RF) is an ensemble of decision trees where each tree is trained on a bootstrapped sample of the data and typically uses a random subset of features at each split. The ensemble's prediction is the majority vote of the trees for classification or the average probability. We denote the RF classifier as $h(X)$, which outputs a probability score $p = h(X)$ representing the confidence that the sample is ransomware (malicious). The RF essentially learns decision rules that combine the presence/absence of certain anomaly patterns and the values of Markov anomaly scores to distinguish ransomware from legitimate software. An example rule a tree might learn: "IF detector_5 is triggered (an anomaly in file encryption API usage) AND the minimum API transition probability < 0.001 (very novel sequence) THEN classify as ransomware." In training, the RF will naturally give more weight to features that yield better splits between ransomware and benign, which provides a form of built-in feature selection. Empirically, Random Forest has been shown to perform well on malware classification tasks, sometimes outperforming deep learning on structured behavioral features, and offers faster training.

To make a binary decision from the RF's output probability, we define a decision threshold τ. The classifier assigns the label "Ransomware" if $p \geq \tau$, and "Benign" otherwise. A common default is $\tau = 0.5$ for balanced errors, but in security, we might choose a lower threshold to favor catching malware (at the expense of some false positives). In a theoretical analysis, one can discuss optimizing τ based on metrics like the F-score. For instance, one could select τ that maximizes the F_2-score (which weights recall higher) to ensure ransomware is caught early. The threshold optimization is part of the Decision Evaluation Layer of the framework.

F. Integration of Components

The theoretical robustness of this framework is rooted in the synergistic integration of its three components. This design creates a multi-dimensional feature space where each component addresses a distinct aspect of anomaly detection, a strategy that aligns with modern calls for layered, hybrid detection frameworks [45], [46]. The Negative Selection Algorithm (NSA) functions as an "immune filter," identifying qualitative anomalies by matching patterns that are definitionally "non-self" and should not exist in benign operation. In parallel, the Markov model acts as a statistical profiler, identifying quantitative anomalies by flagging behavioral sequences that, while not strictly impossible, are highly improbable under the learned model of benign behavior [47]. For instance, the RF can learn to disregard a minor qualitative anomaly flagged by the NSA as a potential false positive if the Markov model assigns a high probability to the sequence, indicating an overall normal behavioral flow. Conversely, a sample that triggers multiple NSA detectors and exhibits a cascade of low-probability transitions presents a feature vector that the RF can decisively classify as malicious.

This integration yields a significant benefit: a high degree of explainability, which is a critical requirement for building trust in automated detection systems [48]. Unlike monolithic "black box" models, which are a noted challenge in security-critical environments [49], a classification event can be deconstructed into its constituent, human-readable parts. The system can provide evidence for why a sample was flagged, detailing both the qualitative detectors triggered by anomalous patterns and the quantitative unlikelihood of its behavioral flow.

This evidence renders the model's internal logic transparent and its outputs auditable [50]. Conceptually, this hybrid model fuses biological anomaly detection with probabilistic behavioral profiling to produce a classification that is both resilient and interpretable. Its resilience stems from this multi-modal design;

the framework does not rely on a single indicator of maliciousness. An adversary must therefore simultaneously evade discrete, pattern-based NSA detectors and mimic the statistical profile of benign behavior, presenting a significantly more complex evasion challenge [34].

Finally, the framework is both modular and extensible. One could incorporate additional features, such as the frequency of file modifications or network connections opened, alongside the API sequence features. While our design uses Random Forest for its proven balance of performance and interpretability, other classifiers like SVM or XGBoost could also be employed.

The core contribution of this study, however, is the integrated NSA-Markov-RF architecture (termed Pipeline 1). To rigorously validate the novel contributions of this specific hybrid design, our proposed methodology (detailed in Section 5) is based on a dual-pipeline comparison. This test will benchmark our probabilistic framework (Pipeline 1) against an alternative hybrid, NSA-N-gram-RF (termed Pipeline 2), to empirically quantify the specific improvements of using a Markov model.

In summary, the proposed theoretical framework is a synergistic integration of three well-established algorithmic pillars: the Negative Selection Algorithm from AIS theory, first-order Markov chains from probability theory, and the Random Forest from ensemble learning. While prior work has validated each component individually for security applications, our primary contribution is the formal design of this cohesive, multi-modal system. This study has articulated this design, demonstrating how the unique, emergent benefits of resilience and interpretability arise from this specific combination.

IV. Proposed Methodology

This study adopts the Design Science Research Methodology (DSRM) framework [20], which is especially well-suited for developing and evaluating novel cybersecurity artefacts. DSRM ensures that artefact creation is grounded in both practical relevance and theoretical rigor. Our ransomware detection framework is constructed and validated in accordance with the following DSRM stages:

- Problem Identification and Motivation: We identified key limitations in current detection approaches, namely, a lack of interpretability (G1) and high false positives (G2) and justified the need for a hybrid, behaviorally aware system (G3).
- Design and Development: We designed a multilayered detection artefact combining a Negative Selection Algorithm (NSA), a probabilistic Markov model, and a Random Forest classifier (Section IV). This artefact is instantiated in two comparative pipelines for controlled evaluation.
- Demonstration: The detection system is implemented and executed on dynamic API trace data collected from both benign applications and ransomware samples.
- Evaluation: We empirically evaluate system performance using standard classification metrics, emphasizing Recall and False Positive Rate (FPR), and using F2-score optimization to reflect real-world detection priorities.
- Communication: This study serves as the primary vehicle for communicating the artefact design, theoretical justification, and validation methodology to the research community.

A. Evaluation Metrics

Model performance will be assessed using a comprehensive suite of standard evaluation metrics: Accuracy, Precision, Recall (Detection Rate), F2-Score, and False Positive Rate (FPR). Emphasis is placed on minimizing FPR. Furthermore, due to the asymmetric cost of errors in ransomware detection were failing to detect an active infection (False Negative) is more damaging than triggering a benign false alarm. This study prioritizes Recall and adopts the F2-Score as the principal optimization target. The F2-Score weights Recall twice as heavily as Precision, thereby aligning metric selection with the risk model inherent to the domain.

To determine the optimal operating point of the classifier, the decision threshold τ will be tuned using a grid search across the interval $[0.1, 0.9]$, with the optimal τ^* defined as the threshold maximizing the F2-Score on a validation subset.

B. Baseline Models and Comparative Hypothesis

To assess the efficacy of the proposed framework, we define a structured comparative evaluation across two distinct pipeline configurations. Both configurations share a common NSA pre-filtering stage and a Random Forest classifier. This strategy isolates the effect of the Markov chain behavioral profiler by comparing it directly with a static, pattern-based encoding scheme (adaptive n-gram).

1) Pipeline 1: NSA - Markov - Random Forest (Full Hybrid Architecture)

This is the proposed end-to-end detection framework, comprising:

- Negative Selection Algorithm (NSA): Eliminates benign "self" behavior patterns from the dynamic trace to generate an anomaly-prone residual signal.
- First-Order Markov Chain Profiler: Models the transition probabilities of the remaining "non-self" API call sequences to quantify behavioral abnormality.
- Random Forest Classifier (RF): Predicts the likelihood of ransomware execution using a feature set that includes Markov likelihood scores and statistical API call metrics.

2) Pipeline 2: NSA-N-gram-Random Forest (Baseline Hybrid)

This configuration retains the NSA filtering stage but replaces the probabilistic Markov model with a flexible n-gram feature extractor:

- Negative Selection Algorithm (NSA): Performs initial filtering to exclude benign behaviors.

- Adaptive N-gram Encoding: Extracts variable-length API call n-gram patterns from the NSA-filtered trace to capture structural nuances in the remaining signal.
- Random Forest Classifier (RF): Classifies the resulting feature vector to predict ransomware presence.

V. Experimental Evaluation

A. Dataset and Sample Collection

To evaluate the proposed ransomware detection framework, a balanced dataset consisting of ransomware and benign software samples was constructed and analyzed in a controlled sandbox environment. The dataset was designed to capture a diverse range of ransomware behaviors while reflecting realistic benign application activity. Ransomware samples were collected from a public malware repository [51]a well-established and actively maintained malware repository widely used in academic research and threat intelligence. Samples span the period from 2015 to June 2023, capturing nearly a decade of ransomware evolution. A structured keyword-based query using the *tag:ransom* was employed to identify binaries explicitly labeled as ransomware. Metadata such as SHA-256 hash, file type, family signature, and classification tags were used for verification and filtering.

To focus specifically on Windows crypto-ransomware, the following selection criteria were applied:

- Only PE32 (exe) Windows executables were retained.
- Locker-type ransomware variants that primarily restrict access through screen locking without performing file encryption were excluded.
- Samples were required to contain the "ransom" or "ransomware" classification tags.

Using this process, 1,200 ransomware samples were initially shortlisted. From this pool, 100 representative samples were selected based on family diversity, file quality, and temporal distribution. These samples span 40 distinct ransomware families, including LockBit, STOP, GandCrab, Dharma, REvil, Chaos, and Babuk, ensuring broad behavioral coverage across both early-generation and modern ransomware variants. Moreover, benign samples consisted of 100 legitimate Windows applications drawn from diverse software categories, including productivity tools, security software, media players, development tools, compression utilities, browsers, and document management software. Examples include VLC, WinRAR, Docker Desktop, Bitdefender, Firefox, and Blender. These applications were selected to represent realistic user activity and to expose the detection system to benign behaviors that may superficially resemble ransomware operations, thereby reducing false positives.

All samples, both malicious and benign, were executed in the same sandbox environment using CAPEv2 (Config And Payload Extraction, version 2) to ensure consistency in behavioral logging. During execution, API call sequences and runtime behaviors were captured and used as input for feature extraction and classification.

B. Dataset Curation and Labelling Considerations

Ransomware family labelling presents inherent challenges due to inconsistent naming conventions across antivirus (AV) vendors. To ensure reliability during sample selection, only binaries classified as ransomware by at least ten distinct AV engines were initially retained. However, because family-level labels often vary across vendors, family attribution was refined using post-execution analysis from the CAPEv2 sandbox, which integrates heuristic detection, YARA-based signatures, and static indicators. These labels were manually cross-validated against outputs from high-confidence AV engines, including Kaspersky, Bitdefender, and ESET. In cases of disagreement, a consistency-based majority approach was used to finalize family assignments.

A further challenge involved distinguishing active ransomware samples from dead or partially executing binaries. Some samples failed to execute payloads or produced empty behavioral reports due to packing, sandbox incompatibilities, or execution failures. To mitigate this, samples were screened for successful execution using multiple indicators, including execution duration, API call density, presence of cryptographic API usage, ransom-note artifacts, and CAPEv2 heuristic scores. Additionally, to address execution failures caused by packing or wrapper layers, samples were subjected to a pre-analysis inspection phase. Binaries identified as packed through heuristic analysis were manually or automatically unpacked prior to sandbox execution, ensuring that the core ransomware payload was behaviorally observable during dynamic analysis. Samples exhibiting no meaningful behavioral activity were excluded.

Although an initial pool of approximately 1,200 ransomware samples was identified, only 100 behaviorally expressive samples were retained for analysis. This curation process ensured that the final dataset prioritized behavioral validity, family diversity, and execution reliability, which is essential for behavior-based ransomware detection models.

C. Experimental Setup and Pipeline Configuration

Based on the evaluation design defined in Section IV, two pipeline configurations were implemented and evaluated under identical experimental conditions. Both pipelines share the same dataset, sandbox execution environment, feature extraction process, and Random Forest classifier configuration. The only difference lies in the behavioral modeling stage, allowing a controlled assessment of the impact of probabilistic Markov modeling.

D. Detection Performance Results

Table II summarizes the detection performance of the two proposed pipelines.

The results demonstrate that both pipelines achieve strong detection performance. Pipeline 2 attains perfect classification accuracy for higher-order n-gram configurations under the evaluated dataset, confirming the effectiveness of NSA-based filtering combined with supervised learning. Pipeline 1 achieves comparably high accuracy and recall while maintaining a consistently low false-positive rate across evaluation folds.

TABLE II. PERFORMANCE COMPARISON OF PROPOSED DETECTION PIPELINES

Pipeline	Accuracy	Recall (TPR)	False Positive Rate (FPR)	Explainability
NSA–Markov–RF (Pipeline 1)	98.5%	≈99.3%	< 1%	High (NSA detectors + probabilistic transitions)
NSA–N-gram–RF (Pipeline 2)	100%	100%	0%	Moderate (feature importance via RF)

E. Contribution of Markov Behavioral Modeling

While both pipelines demonstrate high classification accuracy, Pipeline 1 incorporates a probabilistic behavioral modeling layer that captures sequential API-call likelihoods rather than relying solely on structural pattern frequency. Markov-derived features quantify how likely observed API-call transitions are under a benign behavioral model, enabling the detection system to distinguish between rare but legitimate activity and behavior that is statistically inconsistent with normal execution.

This probabilistic profiling contributes to stable detection performance across diverse behavioral patterns and provides interpretable evidence in the form of low-probability transition sequences. As a result, Pipeline 1 offers enhanced behavioral transparency without sacrificing detection effectiveness.

VI. CONTRIBUTIONS AND DISCUSSION

The proposed NSA–Markov–RF ransomware detection framework makes several important contributions to the field of cybersecurity and malware detection, particularly in the context of behavior-based defenses.

- Novel Hybrid Architecture (Solves G3): We introduce a novel hybrid detection architecture that integrates an Artificial Immune System (AIS) anomaly filter, a probabilistic Markov chain model, and an ensemble machine learning classifier (Random Forest) in one unified framework. This addresses the G3 (AIS-ML) gap by proposing a formal, synergistic integration of these components, which prior works have typically been used only in isolation or in simpler pairwise combinations. Moreover, each component is grounded in a well-established theory. By formally describing the framework and providing equations for key processes such as detector generation and transition probability computation, we ensure the design is transparent and reproducible.
- Improved Explainability (Solves G1): A key contribution is improved explainability, directly addressing the G1 (Explainability) gap. Because the model's decision is based on human-comprehensible components such as "detector X matched" or "transition Y→Z had a low probability", analysts can interpret why an alert was raised, building trust in the system [52].
- Robust and Adaptive Classification: The use of an ensemble classifier (RF) confers robustness through bagging and vote aggregation [32]. Furthermore, the framework is modular and adaptive.
- Addressing False Positives (Solves G2): Our framework explicitly tackles the G2 (False Positive) gap. By using the NSA to proactively filter "self" behavior, the classifier is provided with a cleaner, anomaly-focused feature space. This provides a direct answer to the high FPR noted in prior work, such as the 4.8% FPR in the Markov-RF model by [36].

In sum, our theoretical framework not only proposes a specific solution but also advances discourse on how to design ransomware detection systems that are comprehensive, transparent, and resilient.

A. Discussion of the Theoretical Implications

The framework offers several theoretical advantages and implications that contextualize its value.

- Comparative Analysis: The proposed hybrid model provides a layered, defense-in-depth. More specifically, our validation methodology (Section V) is designed to compare Pipeline 1 (NSA-Markov-RF) against Pipeline 2 (NSA-Adaptive N-gram-RF). While both pipelines benefit from the NSA's 'non-self' filtering, the Markov component (P1) provides a probabilistic, behavioral model, whereas the n-gram component (P2) provides a static, pattern-matching one. This is a crucial distinction: an n-gram model (P2) can be evaded if an adversary uses known malicious patterns in a new or rare order. The Markov model (P1), however, is designed to detect precisely this type of anomaly: a sequence of individually plausible API calls combined in a probabilistically rare way. This multi-faceted approach, blending different AI techniques, is in line with what recent surveys suggest for improving novel threat detection [53].
- Explainability and Analyst Trust: The interpretability of our framework means that in a real-world setting, it could generate not just an alert but also a diagnosis. For instance: "Alert: Suspected Ransomware. Reasoning: Detectors D5 and D7 triggered (unseen file encryption sequence), and sequence transition probability average = 0.002 (below normal threshold)." Such feedback is invaluable to an analyst triaging the alert and directly addresses a major criticism of AI in security, the lack of trust in opaque, "black box" models [54].
- Performance Considerations: A potential concern is computational overhead, especially at runtime. NSA detector generation was performed offline during training. During operation, although checking many detectors can be computationally demanding, the use of short, fixed-length behavioral patterns allows for efficient pattern matching. In practice, this ensures that the system is lightweight and suitable for real-time detection. Markov chain calculations are computationally trivial, and Random Forest classification is fast and parallelizable [55].

B. Limitations and Future Work

While promising, the theoretical framework has limitations that must be acknowledged:

- Dependency on Quality of Training Data: The NSA's effectiveness hinges on a comprehensive representation of "self." If benign training data are not comprehensive, the NSA may generate flawed detectors. This limitation is common to all behavior-based models that must learn a "normal" baseline [56].
- Evasion Tactics: Attackers can try to evade each component by mimicking normal API call transitions to fool the Markov model. However, our framework raises a bar for evasion: an adversary must evade both pattern-based detectors and the probabilistic sequence model simultaneously, which is a significantly more complex challenge [46].
- False Positive Handling: A persistent challenge for anomaly-based detection is the handling of false positives. By design, such systems will inevitably flag benign but unusual activities such as those performed by legitimate compression utilities that deviate from the established "self" profile. While the Random Forest classifier is trained to learn and distinguish these benign anomalies, a practical, operational deployment would almost certainly benefit from a supplemental whitelist of known-safe, signed applications. This measure would further reduce false positive noise and substantially improve security analyst trust in the system.

C. Threat Model and Evasion Considerations

The threat model assumes ransomware adversaries capable of employing code obfuscation, packing, and behavioral variation to evade signature-based and static detection mechanisms. Attackers may attempt to mimic benign API-call sequences or reorder known malicious operations to bypass pattern-based classifiers.

The proposed framework raises the bar for evasion by combining complementary detection mechanisms. NSA-based filtering targets previously unseen behavioral patterns, while the Markov chain profiler detects statistically improbable API-call transitions even when individual calls appear benign. To evade detection, an adversary must simultaneously suppress NSA detector activation and maintain transition probabilities consistent with benign execution profiles, which significantly constrains evasion strategies.

While advanced attackers may attempt to gradually adapt behavior to resemble benign profiles, such mimicry reduces operational efficiency and increases attack complexity. The framework, therefore, provides resilience against common behavioral evasion techniques without relying on static signatures.

D. Computational Feasibility and Runtime Considerations

The proposed framework is designed to be computationally feasible for practical deployment. NSA detector generation is performed offline during the training phase, eliminating runtime overhead associated with detector creation. During detection, NSA-based matching operates on short, fixed-length behavioral patterns, enabling efficient pattern comparison.

The Markov chain component incurs minimal computational cost, as transition probability lookup and aggregation are linear in the length of the observed API-call sequence. Random Forest classification is well-suited to real-time analysis due to its parallelizable structure and fast inference time.

Overall, the framework introduces limited runtime overhead beyond standard dynamic behavior analysis and is suitable for near real-time ransomware detection in sandboxed or endpoint monitoring environments.

E. Future Improvements

This theoretical work can be extended in several ways. One potential improvement is employing a higher-order Markov chain or a Hidden Markov Model (HMM) to capture long-term dependencies. The other is to incorporate Positive Selection (another AIS algorithm) in tandem with NSA to detect both known-malicious "antigen signatures" and unknown "non-self" anomalies, as proposed in other AIS-hybrid research [57]. Additionally, exploring Explainable AI (XAI) tools such as SHAP [58] on the Random Forest could precisely quantify the contribution of each feature to a decision, essentially auditing the model's knowledge.

F. Scientific and Practical Impact

Scientifically, our work bridges multiple research areas: bio-inspired security, probabilistic modelling, and machine learning. Practically, the enormous and growing financial and operational costs of ransomware incidents, which cost organizations an average of $4.75 million per breach (excluding the ransom), create a clear need for such tools[59]. Our framework's emphasis on adaptability and its focus on fundamental behaviors (which persist even as tactics evolve) make it a forward-looking approach. It provides a balanced trade-off between detection power and interpretability, building upon validated components from the literature to create a multilayered detection system.

VII. Conclusion

This study presented a bio-inspired, behavior-based hybrid framework for ransomware detection that integrates a Negative Selection Algorithm, probabilistic Markov chain behavioral modeling, and Random Forest classification. The framework addresses key limitations of existing ransomware detection approaches by combining anomaly filtering, probabilistic sequence analysis, and supervised learning within a transparent and modular architecture.

Experimental evaluation using real-world ransomware and benign software samples demonstrated that both evaluated pipelines achieve strong detection performance. While the NSA–N-gram–RF pipeline attained perfect classification accuracy under selected configurations, the NSA–Markov–RF pipeline provided enhanced behavioral interpretability through probabilistic transition modeling while maintaining a consistently low false-positive rate.

By enabling detection decisions to be explained in terms of anomalous behavioral patterns and statistically rare execution sequences, the proposed framework directly addresses the

explainability and false-positive challenges that limit the operational adoption of machine-learning-based ransomware detection systems. Future work will explore early-stage detection, adaptive learning under concept drift, and extension of the framework to additional malware categories.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Ministry of Education Malaysia (Higher Education) for the financial support of this research. This project is funded under the Fundamental Research Grant Scheme (FRGS). This project is funded under the Fundamental Research Grant Scheme (FRGS) [FRGS/1/2015/ICT01/USIM/02/1] and Universiti Kebangsaan Malaysia.

REFERENCES

[1] Y. O. Okoro, M. Oladeinde, O. J. Akindote, A. O. Adegbite, and T. O. Abrahams, "Digital communication and U.S. economic growth: A comprehensive exploration of technology's impact on economic advancement,"Computer Science and IT Research Journal, vol. 4, no. 3, pp. 351–367, Jan. 2024, doi: 10.51594/csitrj.v4i3.687.

[2] S. Kaggwa, D. I. Akinwolemiwa, S. O. Dawodu, P. U. Uwaoma, O. J. Akindote, and S. O. Eloghosa,, "Digital transformation and economic development: A review of emerging technologies' impact on national economies," World Journal of Advanced Research and Reviews, vol. 20, no. 3, pp. 888–905, Dec. 2023, doi: 10.30574/wjarr.2023.20.3.2541.

[3] B. T. Familoni,, "Cybersecurity challenges in the age of AI: Theoretical approaches and practical solutions," Computer Science & IT Research Journal, vol. 5, no. 3, pp. 703–724, Mar. 2024, doi: 10.51594/csitrj.v5i3.930.

[4] T. Sendjaja, E. Prastiawan, Y. Suryani, and E. Fatmawati, "Cybersecurity In The Digital Age: Developing Robust Strategies To Protect Against Evolving Global Digital Threats And Cyber Attacks," International Journal of Science and Society, vol. 6, no. 1, pp. 1008–1019, 2024, doi: 10.54783/ijsoc.v6i1.1098.

[5] M. Masum, M. Jobair Hossain Faruk, H. Shahriar, K. Qian, D. Lo, and M. Islam Adnan, "Ransomware Classification and Detection With Machine Learning Algorithms," in Proc. 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pp. 316–322, 2022, doi: 10.1109/CCWC54503.2022.9720869.

[6] G. Nagar, "The Evolution of Ransomware: Tactics, Techniques, and Mitigation Strategies," International Journal of Scientific Research and Management (IJSRM), vol. 12, no. 06, pp. 1282–1298, Jun. 2024, doi: 10.18535/ijsrm/v12i06.ec09.

[7] S. , Kamil, H. S. A. S. , Norul, A. , Firdaus, and O. L. Usman, "The Rise of Ransomware: A Review of Attacks, Detection Techniques, and Future Challenges," in 2022 International Conference on Business Analytics for Technology and Security, ICBATS 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICBATS54253.2022.9759000.

[8] R. Agrawal, J. W. Stokes, K. Selvaraj, and M. Marinescu, "Attention in recurrent neural networks for ransomware detection," in Proc. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3222–3226, May 2019, doi: 10.1109/ICASSP.2019.8682899.

[9] S. Poudyal and D. Dasgupta, "AI-Powered Ransomware Detection Framework," in 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1154–1161. doi: 10.1109/SSCI47803.2020.9308387.

[10] E. S. Alomari, R. R. Nuiaa, Z. A. A. Alyasseri, H. J. Mohammed, N. S. Sani, M. I. Esa, and B. A. Musawi, "Malware Detection Using Deep Learning and Correlation-Based Feature Selection," Symmetry (Basel), vol. 15, no. 1, Jan. 2023, doi: 10.3390/sym15010123.

[11] J. Beerman, D. Berent, Z. Falter, and S. Bhunia, "A Review of Colonial Pipeline Ransomware Attack," in Proceedings - 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing Workshops, CCGridW 2023, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 8–15. doi: 10.1109/CCGridW59191.2023.00017.

[12] A. A. M. Ali Alwashali, N. A. A. Rahman, and N. Ismail, "A Survey of Ransomware as a Service (RaaS) and Methods to Mitigate the Attack," in Proceedings - International Conference on Developments in eSystems Engineering, DeSE, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 92–96. doi: 10.1109/DESE54285.2021.9719456.

[13] M. A. , Salah, M. F. , Marhusin, and R. Sulaiman, "A Two-stage Malware Detection Architecture Inspired by Human Immune System," in Cyber Resilience Conference (CRC), IEEE, 2018. doi: 10.1109/CR.2018.8626867.

[14] M. A. , Salah, M. F. , Marhusin, and R. Sulaiman, "Malware Research Directions: A Look into Ransomware," Asian Journal of Information Technology, vol. 16, no. 6, 2017.

[15] S. Gulmez, A. G. Kakisim, and I. Sogukpinar, "XRan: Explainable deep learning-based ransomware detection using dynamic analysis," Computers & Security, vol. 139, Art. no. 103703, Apr. 2024, doi: 10.1016/j.cose.2024.103703.

[16] L. Albshaier, S. Almarri, and M. M. H. Rahman, "Earlier Decision on Detection of Ransomware Identification: A Comprehensive Systematic Literature Review," Aug. 01, 2024, Information, , vol. 15, no. 8, Aug. 2024, doi: 10.3390/info15080484.

[17] A. Alraizza and A. Algarni, "Ransomware Detection Using Machine Learning: A Survey," Sep. 01, 2023, Big Data and Cognitive Computing, vol. 7, no. 3, Sep. 2023, doi: 10.3390/bdcc7030143.

[18] H. J. Chittooparambil, B. Shanmugam, S. Azam, K. Kannoorpatti, M. Jonkman, and G. N. Samy, "A review of ransomware families and detection methods," in Advances in Intelligent Systems and Computing, Springer Verlag, 2019, pp. 588–597, June. 2019. doi: 10.1007/978-3-319-99007-1_55.

[19] P. Sharma and K. Chaudhary, "Bio-Inspired Negative Selection Tested for Ransomware Anomaly Detection," in Computing and Machine Learning, CML 2024, Lecture Notes in Networks and Systems, vol. 856, pp. 237–250, Dec. 2024. doi: https://doi.org/10.1007/978-981-97-7571-2_19.

[20] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," Journal of Management Information Systems, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.

[21] S. H. Kok, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Ransomware, Threat and Detection Techniques: A Review," IJCSNS International Journal of Computer Science and Network Security, vol. 19, no. 2, Feb. 2019.

[22] A. Lumazine, G. Drakos, M. Salvatore, et al., "Ransomware Detection in Network Traffic Using a Hybrid CNN and Isolation Forest Approach." Authorea, Oct. 2024, doi: 10.22541/au.172901014.44599790/v1.

[23] S. Satpathy and P. K. Swain, "Graph-contrast ransomware detection (GCRD) with advanced feature selection and deep learning," Discover Computing, vol. 28, no. 1, Dec. 2025, doi: 10.1007/s10791-025-09651-w.

[24] E. Landril, S. Valente, G. Andersen, and C. Schneider, "Ransomware Detection through Dynamic Behavior-Based Profiling Using Real-Time Crypto-Anomaly Filtering," Control System Engineering, Oct. 2024.

[25] B. Barchuk and K. Volkov, "Antivirus evasion techniques in modern malware," World Journal of Advanced Research and Reviews, vol. 26, no. 2, pp. 3154–3166, May 2025, doi: 10.30574/wjarr.2025.26.2.1966.

[26] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware Detection Issues, Challenges, and Future Directions: A Survey," Applied Sciences, vol. 12, no. 17, Sep. 2022, doi: 10.3390/app12178482.

[27] K. Goeschel, "Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis," in SoutheastCon, IEEE, pp. 1–6, March. 2016 doi: 10.1109/SECON.2016.7506774.

[28] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection," arXiv preprint arXiv:1609.03020, Sep. 2016, doi: 10.48550/arXiv.1609.03020.

[29] N. Scaife, H. Carter, P. Traynor, and K. Butler, "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 303-312, June. 2016, doi: 10.1109/ICDCS.2016.46.

[30] J. A. Herrera-Silva and M. Hernández-Álvarez, "Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms," Sensors, vol. 23, no. 3, Feb. 2023, doi: 10.3390/s23031053.

[31] Y.-C. Wu and Y.-L. Chang, "Ransomware Detection on Linux Using Machine Learning with Random Forest Algorithm," TechRxiv, June. 2024, doi: 10.36227/techrxiv.171778770.06550236/v1.

[32] B. M. Khammas, "Ransomware Detection using Random Forest Technique," ICT Express, vol. 6, no. 4, pp. 325–331, Dec. 2020, doi: 10.1016/j.icte.2020.11.001.

[33] S. H. Kok, A. Abdullah, and N. Z. Jhanjhi, "Early detection of crypto-ransomware using pre-encryption detection algorithm," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 5, pp. 1984–1999, May. 2022, doi: 10.1016/j.jksuci.2020.06.012.

[34] S. Li, Q. Zhou, R. Zhou, and Q. Lv, "Intelligent malware detection based on graph convolutional network," Journal of Supercomputing, vol. 78, no. 3, pp. 4182–4198, Feb. 2022, doi: 10.1007/s11227-021-04020-y.

[35] M. A. Saleh, "A Proactive Approach for Detecting Ransomware based on Hidden Markov Model (HMM)," International Journal of Intelligent Computing Research, vol. 10, no. 3, Sep. 2019.

[36] J. Hwang, J. Kim, S. Lee, and K. Kim, "Two-Stage Ransomware Detection Using Dynamic Analysis and Machine Learning Techniques," Wireless Personal Communications, vol. 112, no. 4, pp. 2597–2609, Jun. 2020, doi: 10.1007/s11277-020-07166-9.

[37] R. A. Mowri, M. Siddula, and K. Roy, "Application of Explainable Machine Learning in Detecting and Classifying Ransomware Families Based on API Call Analysis," arXiv preprint arXiv:2210.11235, Nov. 2022, doi: 10.48550/arXiv.2210.11235.

[38] S. Forrest; A.S. Perelson; L. Allen; R. Cherukuri, "Self-nonself discrimination in a computer," in Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy, , pp. 202-212, May, doi: 10.1109/RISP.1994.296580.

[39] S. A. Hofmeyr, "An Immunological Model of Distributed Detection and Its Application to Computer Security," Ph.D. dissertation, Univ. of New Mexico, Albuquerque, NM, USA, 1999.

[40] H. Alrubayyi, G. Goteng, and M. Jaber, "AIS for Malware Detection in a Realistic IoT System: Challenges and Opportunities," Network, vol. 3, no. 4, pp. 522–537, Dec. 2023, doi: 10.3390/network3040023.

[41] T. Lu, Z. Lu, W. Shunye, and G. Qi, "Ransomware detection based on V-detector negative selection algorithm," in 2017 International conference on security, pattern analysis, and cybernetics (SPAC), IEEE, pp. 531–536, Dec. 2017, doi: 10.1109/SPAC.2017.8304335.

[42] T. Lu, Y. Du, J. Wu, and Y. Bao, "Ransomware detection based on an improved double-layer negative selection algorithm," in Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, Springer, pp. 46–61, March. 2020, doi: 10.1007/978-3-030-43215-7_4.

[43] Y. Feng, C. Liu, and B. Liu, "Poster : A New Approach to Detecting Ransomware with Deception," In 38th IEEE symposium on security and privacy, May. 2017.

[44] A. Gazet, "Comparative analysis of various ransomware virii," Journal in Computer Virology, vol. 6, no. 1, pp. 77–90, Feb. 2010, doi: 10.1007/s11416-008-0092-2.

[45] N. , Maheswaran, S. Bose, G. Logeswari, and T . Anitha., "Hybrid Intrusion Detection System Using Machine Learning Algorithm," in Proceedings of Data Analytics and Management, Springer, pp. 333–346, March. 2023, doi: 10.1007/978-981-19-7615-5_30.

[46] G. Li, S. Wang, Y. Chen, J. Zhou, and Q. Zhao, "A Hybrid Framework for Ransomware Detection Using Deep Learning and Monte Carlo Tree Search," OSFPREPRINTS, 2024.

[47] Q. Al-Na'amneh, A. N. Nasayreh, R. Al Mamlook, H. Gharaibeh, A. M. Alsheyab, and M. Almaiah, "Improving memory malware detection in machine learning with random forest-based feature selection," in Risk Assessment and Countermeasures for Cybersecurity, IGI Global, pp. 96–114, 2024 doi: 10.4018/979-8-3693-2691-6.ch006.

[48] H. Manthena, J. C. Kimmel, M. Abdelsalam, and M. Gupta, "Analyzing and Explaining Black-Box Models for Online Malware Detection," IEEE Access, vol. 11, pp. 25237–25252, March. 2023, doi: 10.1109/ACCESS.2023.3255176.

[49] H. Manthena, S. Shajarian, J. C. Kimmell, M. Abdelsalam, S. Khorsandroo, and M. Gupta, "Explainable Artificial Intelligence (XAI) for Malware Analysis: A Survey of Techniques, Applications, and Open Challenges," IEEE Access, vol. 13, pp. 61611–61640, March. 2025, doi: 10.1109/ACCESS.2025.3555926.

[50] A. I. Udofot, O. M. Oluseyi, and E. B. Edim, "Explainable AI for cyber security. Improving transparency and trust in intrusion detection systems," International Journal of Advances in Engineering and Management, vol. 06, no. 12, pp. 229–240, Dec. 2024, doi: 10.35629/5252-0612229240.

[51] abuse.ch, "MalwareBazaar: A repository of malware samples." Accessed: Dec. 15, 2025. [Online]. Available: https://bazaar.abuse.ch/

[52] U. Ahmed, Z. Jiangbin, A. Almogren, S. Khan, M. T. Sadiq, A. Altameem, and A. U. Rehman, "Explainable AI-based innovative hybrid ensemble model for intrusion detection," Journal of Cloud Computing, vol. 13, no. 150, Oct. 2024, doi: 10.1186/s13677-024-00712-x.

[53] Y. Guo, "A Review of Machine Learning-Based Zero-Day Attack Detection: Challenges and Future Directions," Computer Communications, Vol. 198, pp. 175–185, Jan. 2023, doi: 10.1016/j.comcom.2022.11.001.

[54] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explainable Artificial Intelligence in CyberSecurity: A Survey," IEEE Access, vol. 10, pp. 93575–93600, Sep. 2022, doi: 10.1109/ACCESS.2022.3204171.

[55] M. Al-Omari and Q. A. Al-Haija, "Performance Analysis of Machine Learning-Based Intrusion Detection with Hybrid Feature Selection," Computer Systems Science and Engineering, vol. 48, no. 6, pp. 1537–1555, Sep. 2024, doi: 10.32604/csse.2024.056257.

[56] U. Urooj, B. A. S. Al-Rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions," Applied Sciences (Switzerland), vol. 12, no. 1, Jan. 2022, doi: 10.3390/app12010172.

[57] P. Hońko, "Adaptive Positive-Negative Selection Approach," in Journal of Physics: Conference Series, Institute of Physics Publishing, vol. 1061, no. 1, p. 012020 April. 2018. doi: 10.1088/1742-6596/1061/1/012020.

[58] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Proc. 31st International Conference on Neural Information Processing Systems (NIPS), pp. 4768–4777, 2017.

[59] IBM, "Cost of a Data Breach Report 2025 The AI Oversight Gap think report," 2025. Accessed: Nov. 16, 2025. [Online]. Available: https://www.ibm.com/reports/data-breach.