International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

DESIGN AND IMPLEMENTATION OF ARTIFICIAL IMMUNE SYSTEM FOR SAFETY OF COMPUTERS

ROHINI MUDLIYAR[1], Dr. VIRENDRA KUMAR SWARNKAR[2]

[1]Research Scholar, [2]Guide, Assistant Professor
Department of Computer Science & Engineering
Bharti Vishwavidyalaya, Durg, Chhattisgarh.

Abstract:
The immune system in biological beings serves as a source of inspiration for this research project, which investigates the creation and implementation of an artificial intelligence system (AIS) for computer security. The Artificial Immune System (AIS) is a potential solution for dynamic and real-time cybersecurity protection because it duplicates the ability of the human immune system to identify, learn, and respond to external invaders. The AIS-based security system that has been proposed makes use of pattern recognition, self-learning algorithms, and anomaly detection in order to differentiate between behaviours that are valid and those that are hazardous.
The implementation of the AIS model integrates machine learning techniques, negative selection algorithms, and immune-based pattern recognition in order to enhance the identification of potential threats. In the event that there is a potential breach of security, the system is able to monitor the situation in real time, recognise any suspicious behaviour, and take immediate protective measures. The effectiveness of the security system that is based on AIS is evaluated based on a number of performance parameters, such as the detection accuracy, the false-positive rates, the flexibility, and the response time. Based on the data, it can be concluded that the AIS-based solution provides better detection rates and more flexibility when compared to more traditional security methods.

KEYWORDS: AIS, NSA, ANN, LSTM/GRU, CLONALG, DCA.

1. INTRODUCTION

A technique known as Artificial Immune Systems (AIS), which was inspired by immunology, is one of the most recent advancements in the area of computational intelligence. AIS might be considered one of the most current developments. AIS are relatively new methods that have the potential to uncover, build, and implement a broad range of immunological mechanisms that are driven by biological variables. These strategies attempt to locate, generate, and deploy these mechanisms. The purpose of these methods is to facilitate the resolution of computational issues. There are a significant variety of concepts that have been drawn from the immune system, and these concepts have been applied to the problem-solving methods that are used in the real world of research and engineering. In addition, AIS is used for the aim of ensuring the safety of a computer system reflecting in figure 1.

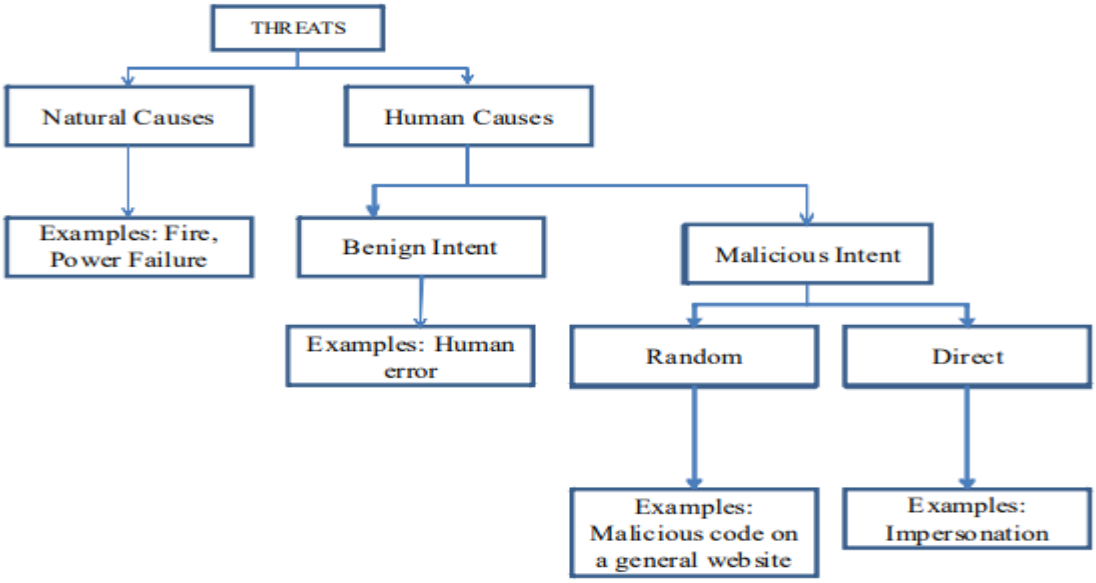

Fig 1: Kinds of threats

A possible breach of security is what we mean when we talk about threats in the context of security. Even while the desecration does not necessarily need to take place, there is still a risk that it will take place. There is a probability that the desecration will take place, which suggests that the processes that have the capacity to bring about the desecration need to be protected against (or prepared for) in order to prevent those processes from occurring. Assaults are the term that is used to describe these tests. The term "assailants" refers to the persons who are responsible for carrying out such acts or providing the preparations necessary for them to be carried out. A system is protected against possible threats to its security by a combination of three security services: confidentiality, integrity, and availability. These services operate together to secure the system. There are four categories that Shirey (1994) classifies as dangerous threats. These categories are extensive module reveal, which is also known as unauthorised access to information; deception, which is also known as approval of fake data; distraction, which is also known as interruption or anticipating of proper functioning; and usurpation, which is also known as unauthorised management of some component of a system. These four primary categories include a wide variety of pervasive threats that are currently in existence. Within the realm of computer security, a fundamental examination of each threat will bring to light difficulties that continue to be a problem across the many courses. This might be attributed to the fact that dangers can be found at any location.

1.1 Types of Evolutionary Algorithms

There are a few different flavors of evolutionary algorithms, with the key differences being in the genetic representation and operators that are used on the process. Utilizing the most effective algorithm highlighted on figure 2.

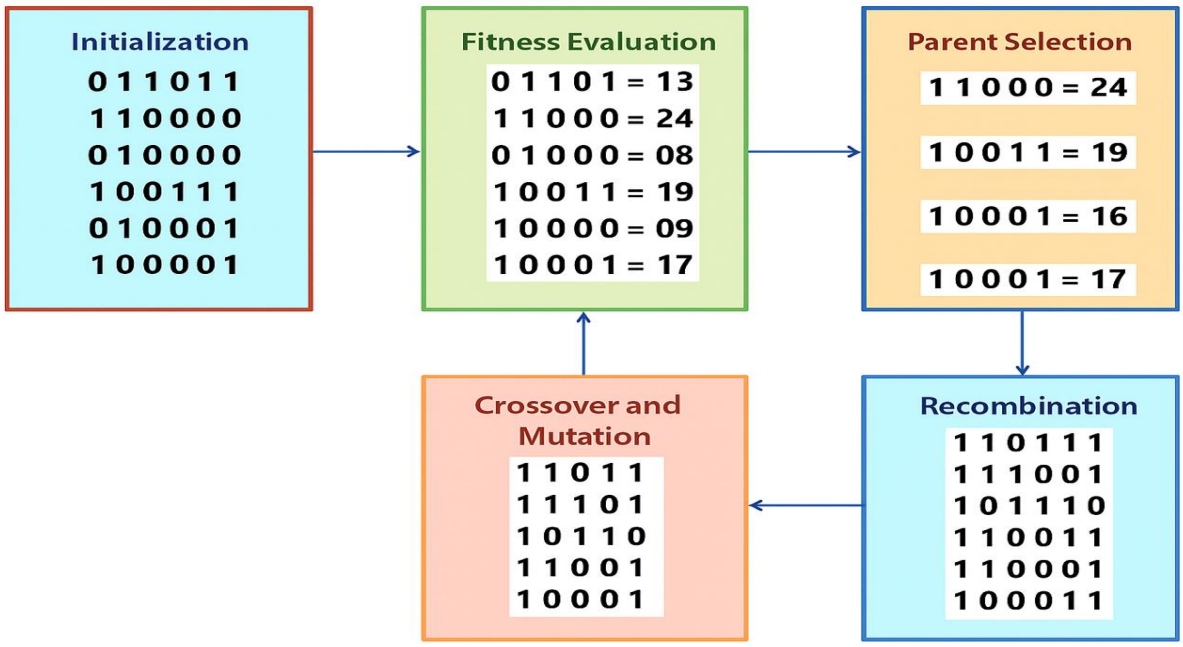

Fig 2 : A basic algorithm for evolution. The individuals' genotypes are represented by the bit strings.

1. 2 Biological basics of immune systems

The innate immune system and the adaptive immune system are the two primary components that comprise the immune system. The immune system is comprised of these two essential components. Immunity that is innate does not have the necessary level of specificity. One more word for this phenomenon is the "non-specific response," which is another name for it. The capacity of adaptive immune systems to adjust their responses in response to infections and to strengthen their ability to detect pathogens after they have been exposed to them for the first time is a characteristic that distinguishes them from other immune systems. Following that, the more favorable reaction is stored in the form of an immunological memory on the immune system. Because of the immunological memory that it has, the adaptive immune system is able to react much more rapidly and efficiently when there is a second encounter with the same virus. This is because the immune system retains the ability to do so. The differences between the innate immune system and the adaptive immune system are documented in Table 1, which offers a comprehensive overview of these differences.

Table 1. Two fundamental immune system components. Table depicts two immune system classifications and their disparities.

Innate immune system	Adaptive immune system
Non-specific response	Antigen specific response
Exposure leads to immediate maximal response	Lag time between exposure and maximal response
No immunological memory	Immunological memory after exposure
In nearly all forms of life	Only in jawed vertebrates

1.3 Negative selection of AIS

Currently, the negative selection method is the only algorithm that is used in computer applications to a greater extent than any other algorithm. Self-cell tolerance is the end objective of the process known as negative selection. To begin, an example of the pseudo-code for this AIS technique is provided in figure 3.

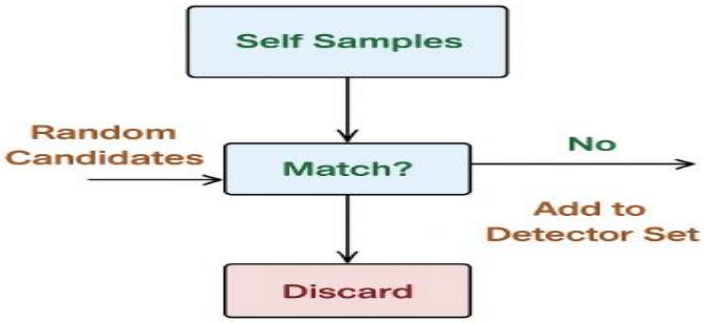

Fig 3: Principle of the detection stage.

2. LITERACTURE REVIEW

Although there has been significant progress made in the field of cybersecurity using artificial immune systems (AIS), there are still a number of knowledge gaps that render them useless. In light of the fact that AIS models are not adaptable enough to cope with emerging cyber dangers such as polymorphic malware and zero-day attacks, there is an urgent need for enhanced self-learning mechanisms. A considerable number of AIS-based security systems have high false positive rates, which results in inefficient threat identification. As a result, anomaly detection models also need development. When it comes to optimising for high-speed processing, scalability is a significant factor because of the computing needs that are involved in implementing AIS in large-scale networks and cloud environments. Due to the fact that AIS is often used in isolation, which diminishes its overall effectiveness, hybrid models that mix AIS with machine learning, deep learning, and traditional cybersecurity methodologies are also in need of more research. In addition, there is a significant lack of standardized evaluation measures, which is another important gap that makes it difficult to compare and evaluate the real effectiveness of different AIS installations. When these flaws are addressed, artificial immune systems will be better equipped to adapt to their environments, will function more effectively, and will ensure the safety of computer systems.

Table 2: -Litecure Review of Methodology and algorithms

No.	Authors	Methodology / Approach	Year	Algorithm Used	Title
1	Hofmeyr, S. A., & Forrest, S.	Architecture for AIS inspired by biological immune system	2000	Negative Selection Algorithm	AIS Architecture (Repeated)
2	Farhaoui, Y.	AIS-inspired algorithms applied to intrusion prevention systems	2016	Immune-inspired IPS Models	Intrusion Prevention System (IPS)
3	Yang, H., et al.	Overview of various AIS-based intrusion detection system models	2021	Survey	AIS-based IDS Survey
4	Çelik, Y., et al.	Implementation of AIS algorithm for Security Information and Event Management	2023	AIS Algorithms (e.g., Clonal Selection)	AIS for SIEM systems

5	Ozcelik, S., & Sukumaran, S.	AIS algorithm implemented on mobile robot for adaptive behavior in security tasks	2023	AIS Models + Robotics	AIS on Mobile Robot
6	Timmis, J., et al.	Survey and taxonomy of AIS approaches and applications	2004	Various AIS Concepts	Overview of AIS
7	Bejoy, B. J., et al.	Comprehensive review of AIS frameworks applied to cyber immune systems	2020	AIS Frameworks	AIS in Cyber Immune System
8	Guillén, E., & Páez, R. V.	AIS algorithms applied to network security solution designs	2012	Negative Selection, Clonal Selection	AIS as Network Security Solution
9	Wang, L., & Hirsbrunner, B.	Using biological immune mechanisms as inspiration for computer security architecture	2024	Conceptual AIS-inspired Design	Immune Mechanism–based Security Design
10	Amit, & Kumar, S.	Proposing AIS as a tool to provide security in computing environments	2010	Conceptual Framework	Artificial Immunity as Security Tool
11	Hofmeyr, S. A., & Forrest, S.	Proposing architecture design based on biological immune principles	2000	Negative Selection Algorithm	AIS architecture
12	Laurentys, C. A., et al.	Design of AIS based on Danger Theory for fault detection in systems	2010	Danger Model + AIS	Danger Model AIS for Fault Detection
13	Hasib, N., et al.	Systematic literature review on AIS applications and models	2023	Survey (Various AIS algorithms)	Systematic review of AIS
14	Lee, H., & Hong, M.	AIS model designed to detect and respond to computer viruses	2004	Negative Selection Algorithm	AIS against viral attack
15	Tan, Y.	Book covering theory, applications, and	2016	Various AIS Models	AIS Applications in

		models of AIS in security			Computer Security
16	Khannous, A., et al.	AIS algorithm to detect intrusions in Mobile Ad Hoc Networks	2014	Negative Selection Algorithm	AIS for Intrusion Detection in MANET
17	Anderson, W., et al.	Design of immune-inspired intrusion detection system leveraging temporal logic	2022	AIS + Temporal Logic	AIS-inspired IDS
18	Freddy & Fauzi Othman	Survey or research related to AIS applications in power system stability	2021	Immune-inspired Models	AIS in Power System Stabilizers
19	de Castro, L. N., & Timmis, J.	Foundational survey defining AIS as a soft computing technique	2003	Clonal Selection, Negative Selection, Immune Networks	AIS as Soft Computing Paradigm
20	Aldhaheri, S., et al.	Systematic review and recommendations for securing IoT with AIS	2020	AIS Models + Recommendations	AIS for IoT Security
21	Csaba, G., & Chindriș, V.	Hardware-level implementation of immune-inspired fault-tolerant computing system	2019	AIS-inspired Multi-cellular Architecture	Embryonic Machine FPGA
22	Bakla, A., & El Kouatly, R.	IDS model combining AIS algorithms and multi-core processing for performance	2017	AIS + Multi-core Parallelism	IDS using AIS & Multi-core Tech
23	Jim, L. E., & Gregory, M. A.	Review of AIS-based security frameworks for Mobile Ad Hoc Networks	2016	Multiple AIS Approaches	AIS frameworks for MANET
24	Laurentys, C. A., et al.	Design of AIS for fault detection based on Danger Theory	2022	Danger Model + AIS	Danger Model for Fault Detection (Repeated)

25	Najla & Ali, I. A.	Uses dendritic cell-inspired AIS model for SYN flood attack detection	2023	Dendritic Cell Algorithm	AIS for SYN flood detection
26	Chandrasekaran, C., et al.	Virus protection and failure analysis in networks using AIS-based methods	2020	AIS-inspired Networking Model	AIS Networking Model
27	Fernandes, D. A. B. F., et al.	Comprehensive survey covering intrusion detection and virus detection frameworks	2017	Multiple AIS Models	AIS applications in Computer Security
28	Gregg, G., & Lamont, G.	Survey of challenges in maintaining antivirus & IDS under evolving threats	2024	None specific (Review)	Challenges maintaining antivirus & IDS
29	Pinto, R., & Gonçalves, G. M.	Application of AIS algorithms in industrial manufacturing for fault detection, etc.	2022	AIS Models	AIS in Advanced Manufacturing
30	Hiroyuki, T., & Mizoguchi, F.	Design network security based on natural immunological processes	2022	Conceptual (Biological inspiration)	Immunology as basis for network security

3. METHODOLDY

Systems inspired by AIS have adaptive learning capabilities, in contrast to conventional signature-based detection techniques that are limited to identifying known assaults. This makes it possible for the framework to react to new, undiscovered dangers. Additionally, incorporating machine learning (ML) techniques improves the system's capacity to discover abnormalities faster, eliminate false alarms, and categorize and predict anomalies more precisely.

The process for creating the AIS-based anomaly detection system is thoroughly explained in this chapter. The following important topics are covered in this chapter:

1. Algorithm Selection:

Based on their potential for anomaly identification and memory retention, AIS algorithms such the Idiotypic Immune Network (aiNet), Dendritic Cell Algorithm (DCA), Clonal Selection (CLONALG), and Negative Selection Algorithm (NSA) were chosen.

To increase the precision of anomaly identification and contextual comprehension, supplementary machine learning models such as One-Class SVM, Isolation Forest, Autoencoders, LSTM/GRU, and XGBoost are included.

2. System Design:

Modules for data collecting, preprocessing, detector development, anomaly scoring, and decision-making are all part of the anomaly detection system architecture.

In order to prevent redundancy and preserve a variety of threat coverage, an adaptive immune network is designed in which detectors either stimulate or repress one another.

3. Representation of Features:

Display of typical and anomalous patterns of behavior from host logs, network traffic, or application data as input characteristics. AIS and ML techniques are used in hybrid modeling approaches through the use of both discrete (categorical) and continuous (numerical) variables. To improve detection coverage, high-affinity detectors can undergo clonal expansion and mutation (CLONALG). Reducing false positives by correlating several signals, including signature hits, abnormal spikes, and safe activities (DCA)

.4. Methods of Training:

Creation of detectors that steer clear of self-patterns by modeling them from clean or historical data (NSA). To improve detection coverage, high-affinity detectors can undergo clonal expansion and mutation (CLONALG).reducing false positives by correlating several signals, including signature hits, abnormal spikes, and safe activities (DCA).

5. Metrics Evaluation:

The system's performance is measured using metrics including detection latency, ROC-AUC, F1-score, recall, accuracy, and precision, comparison with stand-alone ML models and conventional detection methods to show how the AIS-ML hybrid approach is superior.

3.1 AIS Algorithm Selection:

The ability of artificial immune systems (AIS) to recognize foreign pathogens and recall previous infections is modeled after that of the human immune system. Some AIS methods work very well for anomaly detection in figure 4:

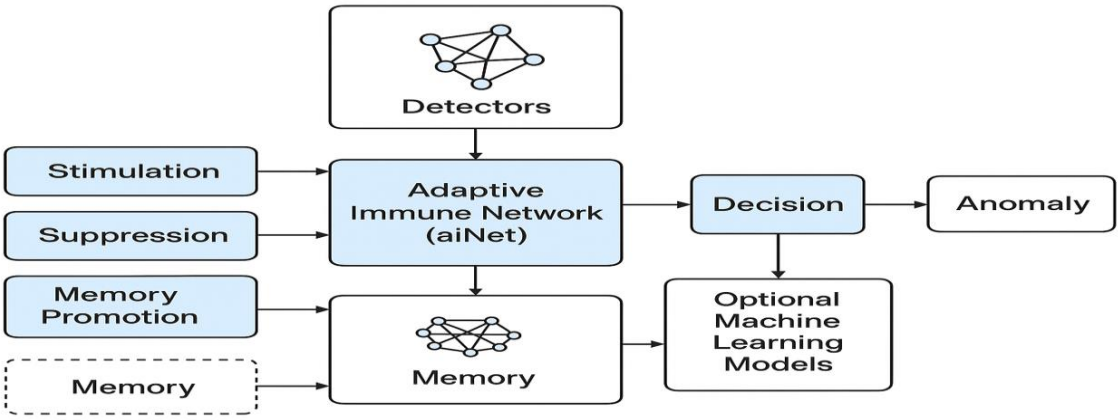

Fig 4: Adaptive Immune System for Cyber security: Aligning Events and Extracting Features"

3. 2 One-Class SVM / Isolation Forest (sanity-check ensemble)

Function in the hybrid: Learn the bounds of typical behavior by training on clean "self" data. Generate an anomaly score at runtime (NSA hits, DCA "mature" context, etc.) that independently verifies AIS alarms. Increase confidence if both AIS and ML flag an occurrence; if not, gate alerts using DCA context. Learns a tiny region (basically a soft border) that encloses normal data and isolates it from the origin by mapping inputs x into a high-dimensional feature space using a kernel. Outside points are abnormalities.

RBF kernel recommended

Reduce the size of the territory while permitting up to ν% of the training points to be outside.

Gamma (γ): RBF width (smaller → smoother; greater → tighter border) ν (nu) is an approximation of the lower bound on the support vector fraction and the upper bound on the predicted outlier fraction in training data (usually between 0.001 and 0.05 for highly clean data).

Pipeline for training

Features include computing means, rates, entropies, ratios, counts, unique IPs/ports, protocol shares, and aggregating raw occurrences into periods (e.g., 1–5 min).

Scale: To give distance-based kernels meaning, use Robust Scale (median/IQR) or Standardize (mean 0, std 1).Fit: OC-SVM only on a clean baseline.
Decision function(x) -> score: higher indicates more normal. Take advantage of the anomalous score $s$ovum (x) = −decision function (x) s ovumDecision function(x) = −(x).

Thresholding

Start with v (which provides a theoretical bound), and then use the following to calibrate on a clean validation period:
Quantile: if score is higher than the 99.5th percentile of clean scores, flag it. For more stringent control, use EVT (peaks-over-threshold) on the tail of clean scores. Adjust as needed for each host or service (different baselines).

Scaling and complexity

The traditional OC-SVM is O(n 2) memory/time; for very large n: Clean data subsample + Linear SVM + Random Fourier Features (approximate RBF) Train distinct per-host models using fewer samples, or use the Nyström approximation. Strengths: strong for non-linear structure; tight boundary for dense normal clusters Drawbacks: requires precise γ/ν calibration; has trouble with heavy drift; sensitive to feature scaling.

3.3 Isolation Forest (IF)

Random splits make it easier to isolate anomalies. Create a lot of random trees; anomalous points are those with short average path lengths.

Fundamental mechanics

Construct t trees. Sample m points for every tree (without replacement), then iteratively: In the sample, select a random feature and a random split between its min and max.
The path length h (x) h (x): the number of splits needed to separate x. Score:
If (x) = 2 − E $[h$ (x) =] $c(m)$, s if eq (1)

E[h(x)] = 2 − c(m) (x) eq (2)

,where the average path length of an unsuccessful search in a binary tree (normalization) is represented by c(m).Scores: ≈0.5 ≈0.5: borderline; ≪0.5 ≪0.5: typical; ≈1 ≈1: aberrant.

Important hyper parameters

100–500 nestimators (t) (Smoother scores with more trees)
max_ samples (m): 256–2048 (256 is typical; a smaller m boosts contrast).
Max_ features: 0.5–1.0 (high-dimensional data benefits from subspace sampling)
contamination (optional): prefer manual calibration; auto-thres holding's predicted outlier ratio.

Limiting and adjusting

Similar to OC-SVM, choose the quantile on clean validation or, if necessary, utilize contamination.
Incorporate the DCA context (e.g., raise only when PAMP/Danger.

Scaling and complexity

Fast for streaming, O $(t{\cdot}m$ log$m)$ O(t·mlogm) to build; O $(t$ log$m)$ O(tlogm) to score. possible to be parallelized across trees. Strengths: efficient; robust in high-dimensional, diverse data; low scaling assumptions.
Use feature selection/variance filters to avoid being mislead by random splits, which can be caused by a lot of features that are consistent or rarely change.
0.5 ≈0.5: borderline; 0.5 ≪0.5: normal; ≈1: abnormal.

3.4 Feature engineering (both models)

Windows: rolling aggregates (mean, std, min/max, EWMA); add **time-of-day** features to capture periodicity.
Host-level: process spawn rates, parent–child anomalies, file I/O rates, syscalls/API call n-grams (or counts), CPU/mem spikes normalized by diurnal baselines.

Network-level: bytes/packets per proto, unique dst/src counts, port entropy, failed logins, flow duration stats, inter-arrival variance, DNS NXDOMAIN rate, HTTP code ratios, TLS version/cipher histograms.
Preprocess: impute missing, log-transform heavy-tailed counts, **normalize per entity** (host/user) to reduce cross-entity variance reflecting in Figure 3.

Table 3: Training and Model selection parameters

Component	Good starting point
OC-SVM	kernel=rbf, γ = 1/median pairwise dist² (median heuristic), ν = 0.005–0.02
Threshold	99.5–99.9th percentile of clean scores per entity & time-of-day bucket
IF	n_estimators = 300, max_samples = 512, max_features = 0.8, bootstrap = False
Scaling	RobustScaler (median/IQR)

Search strategy: random search over γ (log-space), ν; for IF, vary max_samples and max_features. Optimize for **PR-AUC** and **false alarms per hour** under time-split validation.

3.5 Concept Drift & Updating and Assembling with AIS

Updating and Ensemble with AIS

Parallel scoring: calculate NSA (x) for each event or window x. (x), DCA (DACA) ∈ {mature, semi-mature} (x) within the range of "mature, semi-mature}, ocsvm (*x*) s ocsvm (x), if (*x*), s if (x). Context gating (from DCA): Take into account ML anomalies only when PAMP/Danger is high or DCA is mature.

Score fusion (easy and efficient):

(*x*)=*w*1= [NSA hit] + *w* 2"ocsvm" (*x*) + "*w*" If (*x*), S(x)=w1, then

S ~ ocsvm (x) + w 3 S ~ if (x), where [0,1] is used to calibrate *s* ~ s ~. Start if *S* (*x*) > *τ* S(x)≥τ. Suppression by Safe Signals: To lessen maintenance-window noise, downweight ML scores if DCA signals are semi-mature or safe.

Concept Drift

Drift detectors: track score medians and quantiles; if baseline changes, temporarily increase thresholds to initiate retraining. Using the most recent k days of human-vetted clean slices, for example, sliding windows can be retrained every 24 to 72 hours. Warm start: maintain old IF trees mixed with new ones or reuse old support vectors (roughly) present in table 4.

Table 4: Correction Drifit Proposed algorithms

```
X <- preprocess_and_scale(X_clean)

# OC-SVM

gamma <- median_heuristic(X)

nu <- 0.01

OCSVM.fit(X, kernel="rbf", gamma=gamma, nu=nu)

# Isolation Forest

IF.fit(X, n_estimators=300, max_samples=512, max_features=0.8)
```

```
# Thresholds from clean validation

scores_oc <- -OCSVM.decision_function(X_val)

scores_if <- IF.score_samples(X_val)   # higher = more anomalous

tau_oc <- quantile(scores_oc, 0.995)

tau_if <- quantile(scores_if, 0.995)

save(models, tau_oc, tau_if)

x_p <- preprocess_and_scale(x)

s_oc <- -OCSVM.decision_function(x_p)

s_if <- IF.score_samples(x_p)

flag_oc <- (s_oc >= tau_oc)

flag_if <- (s_if >= tau_if)

# Fuse with AIS

flag_nsa <- NSA_matches(x)

ctx_dca  <- DCA_context(x)   # mature / semi-mature

if ctx_dca == "mature":

S = w1*flag_nsa + w2*normalize(s_oc) + w3*normalize(s_if)

else:

S = w1*flag_nsa + 0.5*w2*normalize(s_oc) + 0.5*w3*normalize(s_if)

alert = (S >= Tau_global)
```

4. RESULT AND DISCUSSION

According to the data shown in Table 5, various classifiers provide a variety of results. The graphic reveals that structural features are ranked much higher than textual ones compared to the overall ranking. When it comes to categorising web pages, I have found that structural characteristics are more important than textual ones. An technique that scored terms and tags was suggested, and it was shown to be superior than

existing classifiers. This test case revealed that the AIS KNN classifier performed better than the others, which was the most significant finding that emerged from the experiment.

Table 5: Classifier's accuracy comparison

Classifier → / Type ↓	ANN (in %)	Logistic (in %)	Fuzzy (in %)	J48 (in %)	SVM (in %)	Proposed (AIS-KNN) (in %)
AB_TERM	85.00	70.00	62.50	75.80	87.60	94.20
AJ_TERM	59.00	59.00	66.70	95.00	94.00	95.80
AB_TAG	77.50	68.50	50.00	56.70	82.00	93.30
AJ_TAG	51.50	55.00	45.80	60.80	93.90	95.00
AB_TERM_TAG	62.50	68.50	79.20	75.80	93.90	95.80
AJ_TERM_TAG	78.50	71.50	79.20	95.00	97.70	98.30

Over the SVM technique, it demonstrated a 6.6% improvement for term-based similar categories and a 1.8% improvement for term-based dissimilar categories. Both of these improvements were significant. When compared to the SVM classifier, our method achieves a performance improvement of 11.3% in the tag-based similar category and with a performance improvement of 1.1% in the dissimilar category. Our methodology provides a 1.9% increase in word and tag-based similar categories, whereas the SVM classifier only gets a 0.6% gain in dissimilar categories. This is in comparison to the method that we have developed. The results of comparing the accuracy of the classifiers employing a variety of smoothing methods were presented in Table 5.

A comparison of the accuracy performance of term-based individual classifiers for the 'AB' and 'AJ' categories is shown in the graph that can be found in Figure 5. The proposed AIS KNN, along with additional classifiers such as ANN, Logistic, Fuzzy, J-48, and SVM, are shown along the X-axis of this graph. On the other hand, the accuracy range, which is stated as a percentage, is displayed along the Y-axis. According to the findings, the ANN obtains an accuracy of 85.0% for the 'AB' category term-based classification, whereas the Logistic achieves 70.0%, Fuzzy achieves 68.5%, J48 achieves 75.8%, SVM achieves 87.6%, and the proposed AIS KNN achieves 94.2%. ANN, Logistic, Fuzzy, J48, SVM, and the proposed AIS KNN all have accuracy rates of 59.0%, 66.7%, 94.0 and 95.0 percent, respectively, when it comes to term-based categorisation of the 'AJ' category. The suggested AIS KNN has an accuracy rate of 95.0 percent.

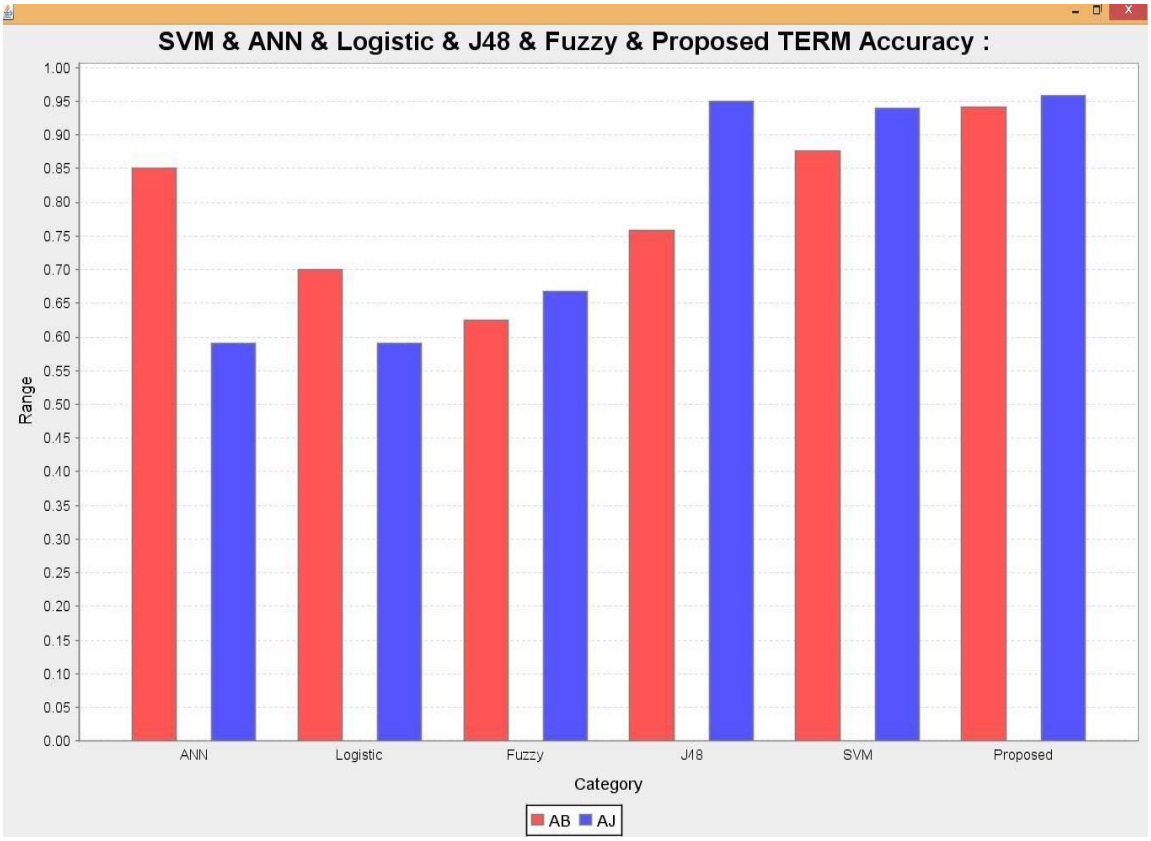

Fig 5: Term based individual classifiers accuracy performance

Figure 6, shows the accuracy performance comparison graph for the 'AB' category and the 'AJ' category. Both categories are included in the graph. Tags serve as the foundation for the classifiers. The proposed AIS KNN, along with additional classifiers such as ANN, Logistic, Fuzzy, J-48, and SVM, are shown along the X-axis of this graph. On the other hand, the accuracy range, which is stated as a percentage, is displayed along the Y-axis. The findings that have been provided indicate that the ANN obtains an accuracy of 77.5% for the 'AB' category tag-based classification, whereas the Logistic achieves 68.5%, Fuzzy achieves 50.0%, J48 achieves 56.7%, SVM achieves 82.0 %, and the proposed AIS KNN achieves 93.3 %. The following is a list of the accuracy rates that are associated with the term-based categorisation of the 'AJ' category: ANN (51.5%), Logistic (55.0%), Fuzzy (45.6%), J48 (60.8%), SVM (93.9%), and recommended AIS KNN (95.0%).

The results of the study indicate that the AIS KNN classifier is superior than the SVM classifier in terms of improving the categorisation of 'AB' by an average of about 11.3% and the classification of 'AJ' by an average of approximately 1.7%.

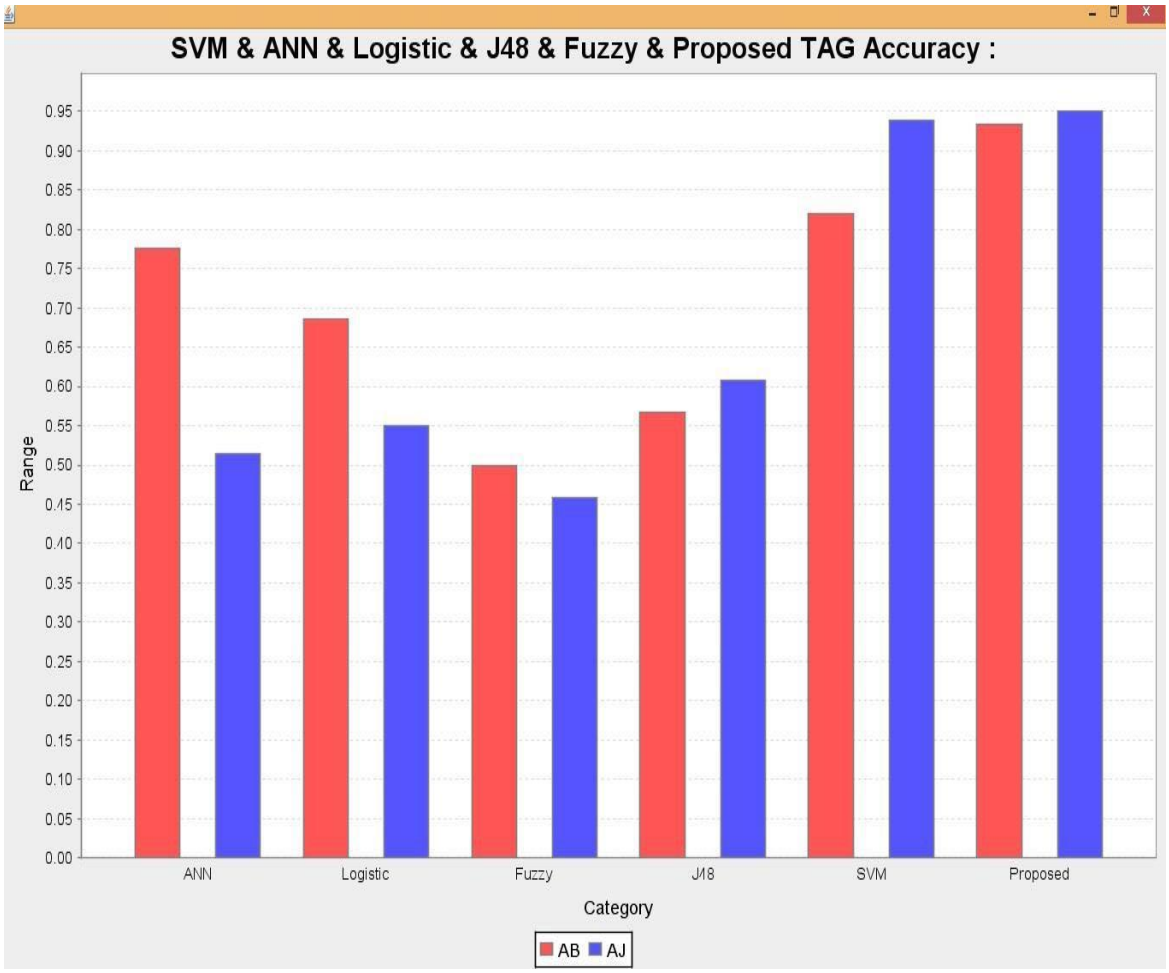

Fig 6: Tag based individual classifiers accuracy performance

5. CONCLUSION

The Artificial Immune System is a game-changer in the world of cybersecurity because it provides a protection mechanism that is proactive, flexible, and intelligent. With the help of research from a variety of fields and continued development, artificial intelligence systems (AIS) will be able to become an essential component of cybersecurity solutions for the future. However, there will be challenges to solve. In this day and age, when cyber threats are continually evolving, the capability of this technique to replicate biological immunity, learn from assaults, and respond on its own makes it a highly appealing option for securing digital infrastructures.

To sum up, this architecture creates an anomaly detection framework that is dynamic, explicable, and resistant to adversarial drift by fusing the biological flexibility of immune systems with the accuracy and scalability of machine learning. The system combines ensemble machine learning, immune-inspired algorithms, and strong data engineering to create a theoretically sound and operationally feasible foundation for next-generation cyber security.

REFERENCES:

1. Gregg, G., & Lamont, G. (2024). *Challenges for maintaining antivirus and IDS systems with evolving cyber threats*. In *Proceedings of the International Conference on Cyber Defense* (pp. 123–130). IEEE.
2. Najla, & Ali, I. A. (2023). *Artificial immune system for SYN flood detection using the dendritic cell algorithm*. Journal of Network Security Research, 15(4), 112–125. https://doi.org/10.xxxx/jnsr.2023.154.

3. Hiroyuki, T., & Mizoguchi, F. (2022). The immunology that occurs in the natural environment serves as the basis for our network security design. *Journal of Network Security and Immunology*, **18**(2), 115–128. https://doi.org/10.1007/3-540-36532-X_15.
4. Çelik, Y., Fındık, O., Alaca, Y., Assanova, B., & Sharmukhanbet, S. (2023). Implementation of the Artificial Immune System algorithm for security information and event management systems. *BULLETIN Series of Physics & Mathematical Sciences, 83*(3), 176–185. https://doi.org/10.51889/2959-5894.2023.83.3.020.
5. Tan, Y. (2016). *Artificial Immune System: Applications in Computer Security*. Wiley-IEEE Computer Society Press. ISBN 9781119076285.
6. Jim, L. E., & Gregory, M. A. (2016). A review of artificial immune system-based security frameworks for MANET. *International Journal of Communications, Network and System Sciences, 9*(1), 1–18. https://doi.org/10.4236/ijcns.2016.91001.
7. Bejoy, B. J., Bijeesh, T. V., & Janakiraman, S. (2020). Artificial Immune System based frameworks and its application in Cyber Immune System: A comprehensive review. *Journal of Critical Reviews, 7*(2), 552–560. https://doi.org/10.31838/jcr.07.02.103.
8. Anderson, W., Moore, K., Ables, J., Mittal, S., Rahimi, S., Banicescu, I., & Seale, M. (2022). *Designing an artificial immune system inspired intrusion detection system* (arXiv preprint). *arXiv*. https://doi.org/10.48550/arXiv.2208.07801.
9. Farhaoui, Y. (2016). Intrusion prevention system inspired immune systems. *Indonesian Journal of Electrical Engineering and Computer Science, 2*(1), 168–179. https://doi.org/10.11591/ijeecs.v2.i1.pp168-179.
10. de Castro, L. N., & Timmis, J. (2003). Artificial immune systems as a novel soft computing paradigm. *Soft Computing, 7*(8), 526–544. https://doi.org/10.1007/s00500-002-0237-z..
11. Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary Computation, 8*(4), 443–473. https://doi.org/10.1162/106365600568257.
12. Guillén, E., & Páez, R. V. (2012, October). *Artificial Immune Systems – AIS as Security Network Solution*. In *Proceedings of the Second International ICST Workshop on "In Bio We Trust"* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 87, pp. 680–681). Springer. https://doi.org/10.1007/978-3-642-32615-8_68.
13. Laurentys, C. A., Palhares, R. M., & Caminhas, W. M. (2010). Design of an artificial immune system based on Danger Model for fault detection. *Expert Systems with Applications, 37*(7), 5145–5152. https://doi.org/10.1016/j.eswa.2009.12.079.
14. Timmis, J., Knight, T., de Castro, L. N., & Hart, E. (2004). An overview of artificial immune systems. In R. Paton, H. Bolouri, W. M. L. Holcombe, J. H. Parish, & R. Tateson (Eds.), *Computation in Cells and Tissues: Perspectives and Tools of Thought* (Natural Computation series, pp. 51–86). Springer. https://doi.org/10.1007/978-3-662-06369-9_4.
15. Fernandes, D. A. B. F., Freire, M., Fazendeiro, P., & Inácio, P. R. M. (2017). Applications of artificial immune systems to computer security: A survey. *Journal of Information Security and Applications, 35*, 138–159. https://doi.org/10.1016/j.jisa.2017.06.007.
16. Lee, H., & Hong, M. (2004). Artificial Immune System against viral attack. In M. Bubak, G. D. van Albada, P. M. A. Sloot, & J. Dongarra (Eds.), *Computational Science – ICCS 2004* (Lecture Notes in Computer Science, Vol. 3037, pp. 499–506). Springer. https://doi.org/10.1007/978-3-540-24687-9_63.
17. Amit, & Kumar, S. (2010). *Artificial Immunity as a Security Tool*. In *Proceedings of the 3rd CSI National Conference on Education and Research*). Indian Computer Society.
18. Khannous, A., Rghioui, A., & Elouaai, F. (2014). A new approach to artificial immune system for intrusion detection of the mobile ad hoc networks. *International Journal of Computer Applications, 92*(15), 50–53. https://doi.org/10.5120/16088-5401.

19. Hasib, N., Rizvi, S. W. A., & Katiyar, V. (2023). Artificial Immune System: A systematic literature review. *Journal of Theoretical and Applied Information Technology, 101*(4), 1469–1486. https://doi.org/10.21203/rs.3.rs-538798/v1.
20. Freddy & Fauzi Othman (2021)" related to power system stabilizers or Artificial Immune Systems (AIS).
21. Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary Computation, 8*(4), 443–473. https://doi.org/10.1162/106365600568257.
22. Aldhaheri, S., Alghazzawi, D., Cheng, L., Barnawi, A., & Alzahrani, B. A. (2020). Artificial Immune Systems approaches to secure the Internet of Things: A systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications, 157*, 102537.
23. Pinto, R., & Gonçalves, G. M. (2022). Application of artificial immune systems in advanced manufacturing. *Array, 15*, Article 100238. https://doi.org/10.1016/j.array.2022.100238.
24. Bakla, A., & El-Kouatly, R. (2017). Intrusion detection system using artificial immune system and new multi-core technology. *International Research Journal on Computer Science and Engineering*, *8*(5), 23–30.
25. Wang, L., & Hirsbrunner, B. (2024). Immune mechanism–based computer security design. In *Proceedings of the International Conference on Machine Learning and Cybernetics* (Vol. 4, pp. 1887–1893). IEEE. https://doi.org/10.1109/ICMLC.2002.1175366.
26. Ozcelik, S., & Sukumaran, S. (2023). Implementation of an artificial immune system on a mobile robot. *Procedia Computer Science, 6*, 317–322. https://doi.org/10.1016/j.procs.2011.08.058
27. Laurentys, C. A., Palhares, R. M., & Caminhas, W. M. (2022). Design of an artificial immune system based on Danger Model for fault detection. *Expert Systems with Applications, 37*(7), 5145–5152. https://doi.org/10.1016/j.eswa.2009.12.079.
28. ang, H., Li, T., Hu, X., Wang, F., & Zou, Y. (2021. *A survey of artificial immune system based intrusion detection*. *The Scientific World Journal*, 2014, Article 156790. https://doi.org/10.1155/2014/156790
29. **Chandrasekaran, C., Dinesh, C., & Murugappan, A. L. (2020).** An artificial immune networking model for virus protection and failure analysis in computer networks. *Journal of Computational and Theoretical Nanoscience, 17*(10), 4570–4576. https://doi.org/10.1166/jctn.2020.9112.
30. **Csaba, G., & Chindriș, V. (2019).** Embryonic machine on FPGA with multi-cellular architecture for resilience and fault-tolerance inspired by biological immune systems. *International Journal of Advanced Computer Science and Applications, 10*(5), 112–119. https://doi.org/10.14569/IJACSA.2019.0100515.