iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** VIII **Month of publication:** August 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.84741

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com*

# AI-Driven Cross-Platform Anomaly Detection and Trust-Aware Cyber Recovery for Mission-Critical Banking Data Infrastructure

Ajaz Ahmed Mohammed[1], Imtiyaz Ahmed Mohammad[2], Mohammad Mustafa[3]
[1]*Advisor, Alinma Isnad (Alinma Bank), Riyadh, Saudi Arabia*
[2]*Software Engineer, Saim Technologies, Cedar Park, Texas, USA*
[3]*Communications Engineer, Contact Center Company (ccc), Riyadh, Saudi Arabia*

***Abstract:*** ***Ransomware and other destructive cyber events increasingly threaten the availability and integrity of mission-critical financial information systems. Modern banking infrastructures are no longer confined to a single storage or compute platform. Enterprise applications increasingly span virtual machines, containerized workloads, object-storage platforms, block and file storage, backup repositories, replication systems, and geographically separated recovery environments. This heterogeneity creates an important security limitation: platform-specific monitoring can identify local anomalies while failing to recognize a coordinated attack whose evidence is distributed across multiple infrastructure layers.***

***This paper proposes CART, a Cross-platform Anomaly and Recovery Trust framework for mission-critical banking data infrastructure. CART combines heterogeneous telemetry from container orchestration platforms, virtualization environments, enterprise storage, object storage, backup systems, replication services, and recovery repositories. A context-aware temporal anomaly-detection layer identifies deviations from workload-specific behavioral baselines and a cross-platform correlation engine combines weak signals across infrastructure domains. Following detection, a Recovery Trust Score (RTS) evaluates candidate recovery points according to integrity evidence, temporal consistency, behavioral exposure, provenance, protection status, and observed anomaly propagation. The framework therefore extends ransomware detection from a binary classification problem— "attack or no attack" —to a detection-to-recovery decision problem: detect the attack, estimate its cross-platform exposure, and identify the most trustworthy recovery state.***

***The proposed methodology addresses several limitations in existing ransomware research, including dependence on single-layer telemetry, insufficient consideration of legitimate high-volume enterprise workloads, concept drift, severe class imbalance, and the assumption that the latest available backup is necessarily the safest recovery point. The study defines a reproducible evaluation protocol using public ransomware datasets, container-security datasets, and controlled synthetic enterprise-storage telemetry. Evaluation will compare conventional threshold detection, individual machine-learning models, platform-specific models, and the proposed cross-platform framework using precision, recall, F1-score, PR-AUC, false-positive rate, mean time to detection, recovery-point selection accuracy, false-safe recovery rate, and estimated recovery-point and recovery-time objectives.***

***The intended contribution is a platform-independent architecture for AI-assisted cyber resilience in heterogeneous financial data infrastructure. Rather than claiming that artificial intelligence can eliminate ransomware, the framework aims to provide measurable improvements in early anomaly detection and evidence-based recovery-point selection.***

***Keywords: ransomware; cyber resilience; anomaly detection; machine learning; banking infrastructure; object storage; Kubernetes; OpenShift; virtualization; backup recovery; disaster recovery; explainable AI; recovery trust; data integrity.***

## I. INTRODUCTION

Financial institutions depend on highly available and highly reliable information infrastructure. Core banking, payment processing, customer-facing applications, analytical platforms, regulatory systems, and supporting services depend on persistent data that must remain available and trustworthy during both normal operations and cyber incidents.

Ransomware creates a particularly difficult form of infrastructure risk because the attack can affect both availability and data integrity. Traditional ransomware scenarios focus on encryption of files or databases; however, contemporary destructive attacks may also involve data exfiltration, deletion, modification of backups, compromise of administrative credentials, and attempts to disrupt recovery mechanisms.

©IJRASET: All Rights are Reserved 

NIST's ransomware risk guidance frames ransomware management as a lifecycle encompassing governance, identification, protection, detection, response, and recovery. NIST's data-integrity guidance further emphasizes that organizations need confidence that recovered data is accurate and safe following destructive events.

This creates a fundamental distinction between detecting an attack and recovering safely from an attack.

A security system may correctly identify ransomware at 12:00, but this does not answer a critical operational question:

### A. *Which Recovery Point can the Organization Trust?*

Consider a heterogeneous banking environment containing:

- OpenShift/Kubernetes workloads;
- virtual machines;
- block and file storage;
- object storage;
- database systems;
- backup repositories;
- storage snapshots;
- replication targets;
- immutable copies; and
- Geographically separated disaster-recovery infrastructure.

An attack beginning at 10:00 may not be detected until 12:00. During those two hours, replication may propagate corrupted data, snapshots may contain partially encrypted information, backup jobs may capture compromised workloads, and object-storage versions may contain both clean and corrupted states.

### B. *Consequently, the most recent backup is not necessarily the safest backup.*

Existing ransomware research has made significant progress in behavioral and machine-learning detection. RanSMAP, for example, introduced an open dataset containing ransomware storage and memory-access patterns and demonstrated the usefulness of low-level infrastructure behavior for ransomware detection. Recent work has also expanded behavioral ransomware datasets considerably; MLRan contains more than 4,800 ransomware samples across 64 families and provides an open reproducibility-oriented pipeline. Other research has investigated concept drift, adversarial ransomware, and machine learning in containerized environments.

However, these research streams are generally focused on individual components of the problem:

- malware or ransomware classification;
- storage-level behavioral detection;
- container anomaly detection;
- infrastructure attack-path analysis; or
- recovery and backup integrity.

The problem addressed by this paper is the integration of these dimensions into one decision framework.

We propose that ransomware resilience in heterogeneous financial infrastructure should be treated as a two-stage intelligence problem:

### C. *Rather than Simply*

The proposed framework, CART, therefore, has two central objectives:

- Detect coordinated anomalous behavior across heterogeneous infrastructure layers; and
- Evaluate recovery points and recommend the recovery state with the highest evidence-based trust.

The principal research question is:

Can cross-platform temporal correlation of infrastructure behavior improve ransomware detection and enable more reliable selection of trustworthy recovery points in heterogeneous mission-critical banking environments?

## II. RESEARCH CONTRIBUTIONS

This study makes five intended contributions.

Contribution 1: Cross-platform anomaly model

We propose a platform-independent telemetry representation capable of incorporating signals from containers, virtual machines, storage, object storage, backup, and replication systems.

Contribution 2: Context-aware anomaly detection

Rather than treating high resource consumption as inherently malicious, the framework evaluates behavior relative to workload and operational context.

Contribution 3: Cross-platform correlation

The framework combines weak signals distributed across multiple infrastructure layers rather than relying on one platform-specific indicator.

Contribution 4: Recovery Trust Score

We introduce a formal score for evaluating candidate recovery points according to integrity, temporal, behavioral, provenance, and exposure evidence.

Contribution 5: Detection-to-recovery evaluation

The research evaluates not only ransomware classification performance but also early detection and recovery-point selection.

## III. BACKGROUND AND RELATED WORK

### *A. Ransomware Detection*

Machine-learning-based ransomware detection has become an important research area because conventional signatures may fail against previously unseen or rapidly evolving variants. Recent research has explored static features, dynamic behavioral features, memory behavior, API calls, storage activity, and system-level telemetry.

MLRan demonstrates the scale now possible in behavioral ransomware datasets, using thousands of samples and multiple ransomware families. Its use of explainability techniques such as SHAP and LIME demonstrates the growing importance of understanding which behavioral characteristics drive classification decisions.

However, malware-level behavior does not necessarily represent the operational context of a banking infrastructure. A production database backup may generate substantial disk activity, while OpenShift deployment may create abrupt changes in network, CPU, memory, and storage utilization.

Therefore, a useful enterprise detector must distinguish:

From and the distinction must be context dependent.

### *B. Storage-Based Ransomware Detection*

Storage behavior provides a valuable additional detection layer because ransomware encryption directly changes data-access behavior.

RanSMAP demonstrated the value of low-level storage and memory access patterns for ransomware detection. Such approaches are particularly valuable when host-level detection mechanisms are compromised or bypassed.

However, storage-only detection can suffer from ambiguity.

Examples of legitimate high-volume activity include:

- database backup;
- database restore;
- data migration;
- ETL processing;
- replication resynchronization;
- large-scale object operations;
- disaster-recovery testing;
- application deployment; and
- batch processing.

A high write rate therefore cannot by itself constitute evidence of ransomware.

The proposed framework consequently treats storage behavior as one component of a larger contextual feature space.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

### C. Concept Drift

Ransomware behavior changes over time. Machine-learning models trained on historical ransomware families may encounter future variants whose statistical behavior differs from the training distribution.

Fernando and Komninos proposed FeSAD to address ransomware concept drift using feature selection and drift detection/adaptation mechanisms.

Concept drift is particularly relevant to the proposed framework because infrastructure behavior itself also changes.

For example:

even when no attack exists.

A production system may undergo:

- application upgrades;
- infrastructure migrations;
- container-platform upgrades;
- storage expansion;
- new backup policies;
- new workloads; and
- changes in replication topology.

CART therefore incorporates drift detection into the proposed architecture rather than assuming a permanently stationary environment.

### D. Containerized Infrastructure

Modern banking workloads increasingly use container orchestration platforms. Kubernetes provides observability through metrics and logs, including application, system-component, and audit information.

Container environments create a different anomaly-detection problem from traditional virtual machines because:

- pods are dynamically created and destroyed;
- workloads scale horizontally;
- deployments change rapidly;
- service-to-service communication changes;
- ephemeral storage may appear and disappear;
- microservices generate distributed telemetry.

Recent research has investigated machine-learning intrusion detection for containerized services using system-call information. Other research has examined dynamic attack graphs and risk assessment for container environments.

These studies demonstrate that container telemetry can provide useful security information, but they do not directly solve the problem of correlating container behavior with downstream storage, backup, object-storage, and recovery-layer evidence.

### E. Backup and Recovery

Backup systems are a critical component of ransomware resilience. NIST SP 1800-11 emphasizes that organizations must be able to identify altered data and determine the correct backup version free of corrupted data.

OpenShift also provides documented backup and recovery mechanisms for cluster state and applications.

The central observation underlying this paper is:

The existence of a backup does not establish the integrity of the backup.

A recovery point may be technically available but operationally unsafe.

Therefore, recovery-point selection should be treated as an evidence-evaluation problem.

## IV. PROBLEM FORMULATION

Let an enterprise environment contain infrastructure domains:

where, for example:

For each domain , the system observes a time-series telemetry vector:

The complete infrastructure observation at time is:

The detection problem is to estimate:

where indicates whether malicious activity is occurring and represents the temporal observation window.
The recovery problem is different.
Let:
represent candidate recovery points.
The framework must estimate:

where represents all available evidence concerning recovery point .
The overall decision problem therefore becomes:
subject to operational constraints such as:
and:

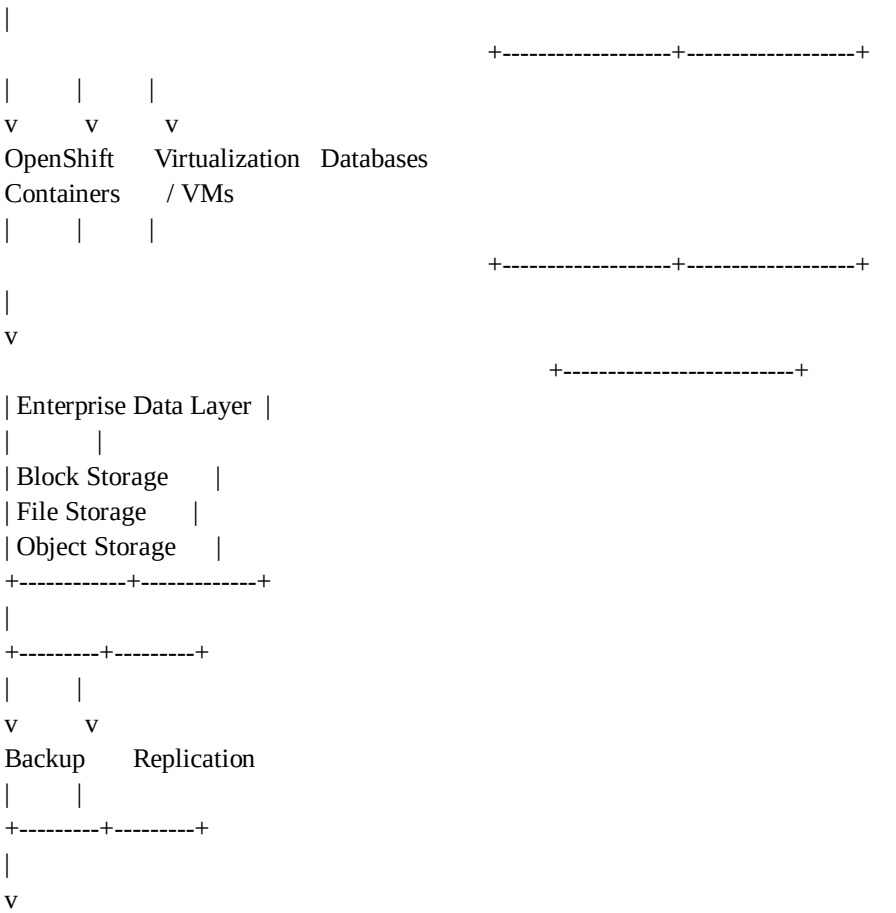
## V. PROPOSED CART FRAMEWORK

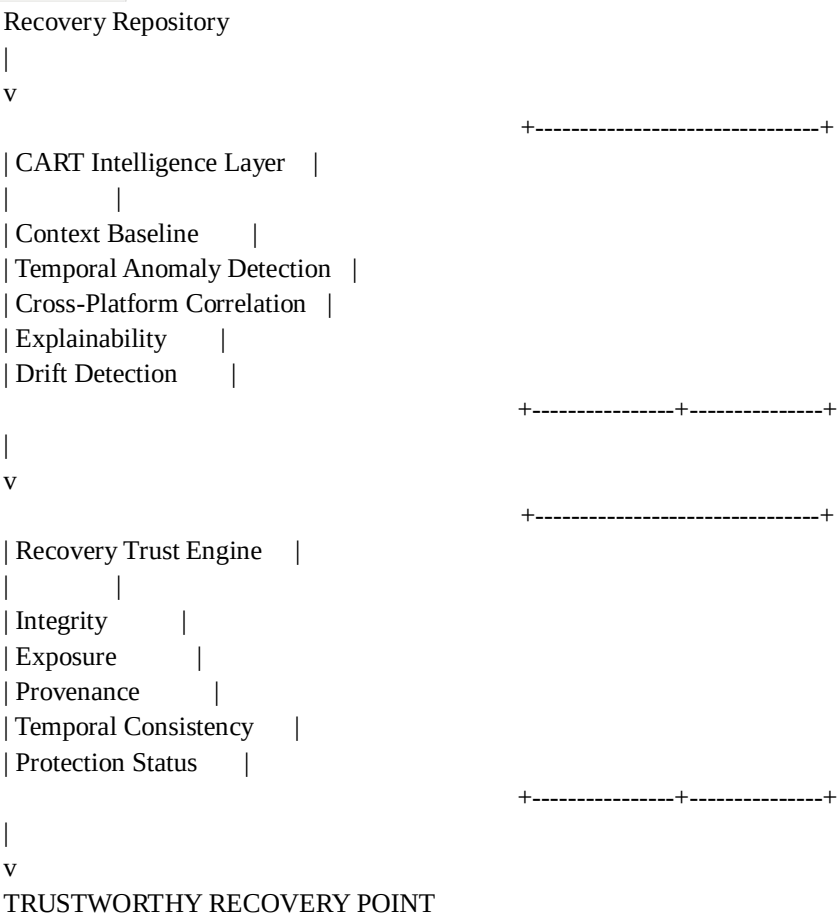
### A. *Architecture*

CART consists of six logical layers:

*1)* telemetry collection;
*2)* normalization and temporal alignment;
*3)* context-aware anomaly detection;
*4)* cross-platform correlation;
*5)* recovery trust assessment; and
*6)* decision support.

Figure 1. Proposed CART architecture






## VI. TELEMETRY MODEL

The framework intentionally avoids dependence on one vendor's APIs.

### *A. Container Telemetry*

Potential features include:

- pod creation/deletion rate;
- restart frequency;
- CPU utilization;
- memory utilization;
- network traffic;
- filesystem activity;
- system-call characteristics;
- container image changes;
- namespace activity;
- Kubernetes/OpenShift audit events.

Kubernetes observability documentation identifies logs, metrics, and audit-related information as important sources of operational and security visibility.

### *B. Virtualization telemetry*

Potential features include:

- VM CPU utilization;
- memory utilization;
- disk I/O;

- network traffic;
- virtual-disk write rate;
- snapshot creation;
- snapshot deletion;
- VM lifecycle events;
- guest-level activity where available.

### *C. Storage Telemetry*

Potential features include:

Additional data-behavior features include:

- file modification rate;
- rename rate;
- extension-change rate;
- entropy variation;
- changed-data percentage;
- snapshot growth.

### *D. Object-storage telemetry*

Object storage introduces a different behavioral vocabulary:

- PUT rate;
- GET rate;
- DELETE rate;
- LIST rate;
- object-version creation;
- version deletion;
- bucket-policy changes;
- access-key anomalies;
- object-size distribution;
- unusual prefix activity;
- replication changes;
- object-lock state changes.

This is important because destructive activity against object storage may not appear as a traditional filesystem encryption pattern.

### *E. Backup telemetry*

Potential features include:

- backup job success/failure;
- backup size deviation;
- changed-block percentage;
- backup duration;
- unusual deletion attempts;
- retention-policy modifications;
- immutable-copy status;
- backup repository activity;
- recovery-test results.

### *F. Replication Telemetry*

Potential features include:

- replication lag;
- replication throughput;
- changed-data volume;

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

- synchronization failures;
- resynchronization events;
- replication-policy changes;
- abnormal directional changes.

## VII. CONTEXT-AWARE BEHAVIORAL BASELINE

A major design principle of CART is that there is no universal definition of “normal.”
For a workload , the baseline is:
where:

- = workload characteristics;
- = temporal context;
- = operational context.

Temporal context may include:

- hour of day;
- day of week;
- month-end period;
- scheduled backup window;
- scheduled maintenance window.

Operational context may include:

- application deployment;
- backup;
- migration;
- disaster-recovery test;
- storage expansion.

The anomaly score is therefore:
rather than simply:
This distinction is fundamental to reducing false positives in enterprise banking environments.

## VIII. HYBRID AI DETECTION ENGINE

CART uses three complementary intelligence mechanisms.

### *A. Unsupervised anomaly detection*

An autoencoder or equivalent anomaly-detection model learns the representation of normal behavior.
Given input:
the reconstruction error is:
A high reconstruction error indicates deviation from learned normal behavior.

### *B. Supervised classification*

Supervised models are trained using labeled benign and ransomware/anomalous observations.
Candidate models include:

- Random Forest;
- XGBoost;
- LightGBM;
- temporal neural networks;
- LSTM;
- temporal convolutional networks.

The research should compare these rather than assuming that a deep-learning model is automatically superior.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

*C. Temporal modeling*

Ransomware is a behavioral process rather than a single event.
Therefore:
is more informative than alone.
The temporal model learns transitions such as:
rather than simply detecting a single high-value metric.

## IX. CROSS-PLATFORM CORRELATION ENGINE

This is a central contribution of CART.
Suppose:
for container anomaly,
for virtualization anomaly,
for storage anomaly,
for object-storage anomaly, and
for backup anomaly.
The framework does not simply average the scores.
Instead, it evaluates temporal and causal relationships.
Let:
where:

- represents temporal relationships;
- represents infrastructure dependency relationships.

A correlated anomaly occurring within a plausible propagation window receives greater weight than independent anomalies occurring at unrelated times.

## X. INFRASTRUCTURE DEPENDENCY GRAPH

The system maintains an abstract dependency graph:
where:

- = infrastructure entities;
- = dependency or data-flow relationships.

For example:

OpenShift Application
|
v
Database
|
v
Block Storage
|
+------> Snapshot
|
+------> Replication
|
v
Backup
|
v
Object Repository

If an anomaly occurs at one node, the framework evaluates whether related downstream anomalies are consistent with propagation.
This is complementary to container risk-analysis approaches such as CORAL, which use attack graphs to evaluate dynamic container attack paths [11].

CART instead focuses on behavioral evidence and data-protection consequences.

## XI. RECOVERY TRUST SCORE

### A. *Motivation*

After detection, candidate recovery points must be evaluated.
Let:

be available recovery points.
For each recovery point , define:

- : integrity confidence;
- : temporal consistency;
- : behavioral consistency;
- : provenance confidence;
- : malware/anomaly-free confidence;
- : attack exposure.

The Recovery Trust Score is:
subject to:

The weights should not be selected arbitrarily in the final study. They should be determined through either:

- expert elicitation;
- optimization on validation data;
- Bayesian calibration; or
- sensitivity analysis.

## XII. RECOVERY EXPOSURE WINDOW

A recovery point should be penalized if it falls inside a period in which the ransomware is believed to have propagated.
Define:
For recovery point at time :
A recovery point generated after strong evidence of compromise should receive a higher exposure penalty.
This allows the framework to distinguish:
latest backup
from:
latest trustworthy backup.

## XIII. RECOVERY TRUST DECISION

The recommended recovery point is:
subject to:
and operational requirements:
If no recovery point satisfies the minimum trust threshold:
the framework should not recommend automatic recovery.
Instead it should produce:
No sufficiently trusted recovery point identified — manual forensic validation required.
This safety condition is essential.
An AI system should not autonomously restore a potentially compromised backup merely because it is the highest-scoring candidate.

## XIV. EXPLAINABLE AI

CART should provide an explanation for both detection and recovery decisions.
For ransomware detection:
High anomaly confidence due to simultaneous increases in storage write activity, object DELETE operations, backup change rate, and container filesystem activity.

©IJRASET: All Rights are Reserved 

For recovery:
Recovery point R3 selected because it predates the estimated propagation window, shows normal behavioral characteristics, has verified protection status, and has lower exposure than later recovery points.
SHAP or an equivalent model-agnostic explanation method can be used for feature attribution.
The explanation itself should be treated as decision support, not proof of causality.

## XV. CONCEPT DRIFT DETECTION

Because both ransomware and enterprise workloads evolve, CART incorporates drift monitoring.
Let:
represent the observed distribution at time .
Concept drift occurs when:
where is a statistical divergence measure and is a calibrated threshold.
Potential drift measures include:

- Population Stability Index;
- Jensen-Shannon divergence;
- Wasserstein distance;
- Kolmogorov-Smirnov statistics.

If significant drift persists, the system should enter:
Model Review Mode
rather than silently continuing to trust an increasingly outdated classifier.
This directly addresses the concern identified in prior ransomware concept-drift research.

## XVI. EXPERIMENTAL METHODOLOGY

### *A. Research objectives*

The experiments will test four primary hypotheses.
H1
Cross-platform telemetry improves ransomware detection compared with platform-specific telemetry.
H2
Context-aware temporal modeling reduces false positives caused by legitimate high-volume enterprise operations.
H3
Cross-platform correlation detects coordinated attacks earlier than independent platform detectors.
H4
Recovery Trust Score selection identifies trustworthy recovery points more accurately than latest-backup and rule-based recovery strategies.

## XVII. DATASETS

### *A. Ransomware Behavioral Data*

The study should use public datasets where licensing permits research use.
RanSMAP provides ransomware storage and memory-access patterns and is particularly relevant to the storage dimension [3].
MLRan provides a broader behavioral ransomware corpus containing more than 4,800 samples across 64 families [4].
The final manuscript should record:

- dataset version;
- acquisition date;
- licensing terms;
- preprocessing steps;
- ransomware families;
- benign samples;
- train/test partition;
- feature count.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

*B. Container telemetry*

Container experiments should use a controlled Kubernetes/OpenShift-compatible laboratory environment.
Potential telemetry includes:

- pod lifecycle;
- CPU;
- memory;
- network;
- filesystem;
- system calls where safely collected;
- audit events;
- application logs.

No production banking data should be used.

*C. Synthetic enterprise-storage telemetry*

Because public ransomware datasets do not fully represent heterogeneous banking infrastructure, controlled synthetic telemetry should be generated.
Scenarios should include:
Normal

- steady application workload;
- scheduled backup;
- database batch processing;
- ETL;
- storage migration;
- replication resynchronization;
- OpenShift deployment;
- object-storage bulk upload;
- disaster-recovery testing.

Malicious

- rapid encryption-like write behavior;
- slow encryption;
- mass object deletion;
- snapshot deletion attempts;
- backup repository manipulation;
- simultaneous multi-platform anomalous activity;
- staged attack propagation.

## XVIII. TRAIN/TEST STRATEGY

Random sample splitting is insufficient because it can produce leakage between related ransomware variants.
The study should therefore use:
Family-separated evaluation
Train on ransomware families:
and test on:
where:
This evaluates generalization to unseen families.
Temporal evaluation
Train on older observations and test on later observations.
This evaluates concept drift.
Cross-platform evaluation
Train with one combination of telemetry sources and evaluate with another to determine whether the framework remains useful when some telemetry is unavailable.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com*

## XIX. BASELINE METHODS

The proposed CART framework should be compared against:
Baseline 1
Static threshold-based detection.
Baseline 2
Storage-only machine-learning detector.
Baseline 3
Container-only anomaly detector.
Baseline 4
Backup-only anomaly detector.
Baseline 5
Random Forest.
Baseline 6
XGBoost.
Baseline 7
Temporal neural model.
Baseline 8
Independent platform decisions without cross-platform correlation.
Proposed
CART.
This comparison is important because otherwise an improvement cannot be attributed to the proposed architecture.

## XX. EVALUATION METRICS

### *A. Classification*

Additional measures:

- ROC-AUC;
- PR-AUC;
- false-positive rate;
- false-negative rate;
- balanced accuracy.

PR-AUC should receive particular attention because real-world security environments are highly imbalanced.

## XXI. MEAN TIME TO DETECTION

Define:

where:

- = attack start;
- = detection time.

Lower MTTD is preferable.
The experiment should report confidence intervals rather than a single mean whenever sufficient repetitions are available.

## XXII. RECOVERY POINT SELECTION ACCURACY

Define:
where represents incidents in which the system selected a recovery point validated as safe according to the experimental ground truth.

## XXIII. FALSE-SAFE RECOVERY RATE

This is one of the most important, proposed metrics.
A false-safe decision is a situation where the system recommends a recovery point that contains compromised or corrupted data.
For a banking recovery system, minimizing this metric may be more important than maximizing conventional classification accuracy.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

## XXIV. RECOVERY-POINT TRUST EXPERIMENT

Each scenario should generate several candidate recovery points.
Example:

R1 ——— R2 ——— R3 ——— R4 ——— R5

^
Attack begins
The system receives all recovery points but does not receive the ground-truth answer.
It calculates:

and selects:

The ground-truth laboratory environment then determines whether was actually clean.

## XXV. ABLATION STUDY

A high-quality journal paper should demonstrate which components actually matter.
We therefore propose:
Model A
Storage only.
Model B
Storage + object storage.
Model C
Storage + object storage + backup.
Model D
All infrastructure telemetry without temporal correlation.
Model E
All telemetry + temporal correlation.
Model F
All telemetry + temporal correlation + context.
Model G
Full CART.
The difference between these configurations will reveal whether the proposed architecture provides meaningful incremental value.

## XXVI. SENSITIVITY ANALYSIS

The Recovery Trust Score depends on multiple factors.
Therefore, vary:
within defined ranges.
The objective is to determine whether recovery decisions remain stable under reasonable weight changes.
A robust system should not change its recommended recovery point because one weight changed marginally.

## XXVII. STATISTICAL ANALYSIS

Each experimental scenario should be repeated sufficiently to estimate variability.
Results should report:

- mean;
- median where appropriate;
- standard deviation;
- 95% confidence interval.

For paired model comparisons, appropriate statistical tests should be selected based on the distribution and experimental design.
The paper should avoid reporting statistical significance solely because a p-value is below an arbitrary threshold; effect size and practical significance should also be reported.

## XXVIII. EXPECTED RESULTS REPORTING

Important: this section must be populated only after experiments are performed.
The final paper should contain a table similar to:

| Model | Precision | Recall | F1 | PR-AUC | FPR | MTTD |
|---|---|---|---|---|---|---|
| Threshold | — | — | — | — | — | — |
| Storage-only ML | — | — | — | — | — | — |
| XGBoost | — | — | — | — | — | — |
| Temporal model | — | — | — | — | — | — |
| CART | — | — | — | — | — | — |

The recovery experiment should report:

| Method | Recovery Selection Accuracy | False-Safe Rate | RPO | Estimated RTO |
|---|---|---|---|---|
| Latest backup | — | — | — | — |
| Rule-based | — | — | — | — |
| CART RTS | — | — | — | — |

No values should be inserted until the experiments produce them.

## XXIX. SECURITY AND SAFETY CONSIDERATIONS

The experimental environment must be isolated from production networks.
No destructive malware should be executed against:

- production systems;
- corporate storage;
- real banking data;
- customer information;
- operational backup repositories.

Ransomware behavior should be reproduced using controlled laboratory workloads or appropriately contained research datasets.
The study should not disclose:

- production topology;
- real bank names;
- customer information;
- credentials;
- internal IP addresses;
- proprietary configuration;
- security-sensitive architecture.

If organization-specific telemetry is eventually used, appropriate authorization, anonymization, data-sharing agreements, and security review must be completed before publication.

## XXX. REPRODUCIBILITY

To support reproducibility, the final study should publish, where legally possible:

- feature definitions;
- preprocessing scripts;
- model configurations;

- train/test split methodology;
- synthetic workload generator;
- evaluation scripts;
- model hyperparameters;
- experiment seeds;
- aggregate results.

The study should not publish confidential banking telemetry.
Public ransomware datasets should be cited according to their licensing requirements.

## XXXI. THREATS TO VALIDITY

*1)* Dataset validity: Public ransomware datasets may not perfectly represent modern enterprise banking environments.
*2)* Synthetic telemetry: Synthetic storage telemetry may fail to reproduce every complexity of real production systems.
*3)* Concept drift: Even a model that performs well during evaluation may degrade as ransomware and infrastructure behavior evolve.
*4)* Vendor heterogeneity: Different storage, object-storage, backup, virtualization, and container platforms expose different telemetry.
*5)* Ground truth: Determining whether a recovery point is clean may itself require forensic validation.
*6)* Automated recovery risk: A high model confidence score cannot guarantee that recovery is safe.

For this reason, CART should support configurable human approval before restoration.

## XXXII. DISCUSSION

The proposed framework changes the conceptual model of ransomware resilience.
Traditional security systems tend to produce:
CART proposes:
This distinction is important because ransomware recovery is fundamentally a data-integrity problem.
NIST's recovery guidance emphasizes the need to determine the correct backup version and restore trustworthy data [2]. CART operationalizes that principle as a machine-assisted evidence-ranking problem.
The cross-platform design is also important because modern infrastructure is heterogeneous.
A container anomaly by itself may be benign.
A storage anomaly by itself may be caused by backup.
An object-storage deletion burst may be part of a legitimate lifecycle process.
A backup anomaly may be caused by a failed job.
But simultaneous, temporally aligned anomalies across multiple dependent systems can provide stronger evidence.
This creates an opportunity for cross-domain correlation to reduce both false positives and detection latency.

## XXXIII. PRACTICAL BANKING APPLICATION

The framework is particularly suited to environments containing:

*1)* core banking applications;
*2)* payment systems;
*3)* customer-data platforms;
*4)* financial databases;
*5)* OpenShift/Kubernetes workloads;
*6)* virtualized applications;
*7)* enterprise storage;
*8)* object-storage repositories;
*9)* centralized backup infrastructure;
*10)* disaster-recovery sites.

The framework is not tied to a specific vendor.
Instead, vendor-specific telemetry should be normalized into a common semantic model.
This allows the research to remain relevant across different infrastructure architectures.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

## XXXIV. RELATIONSHIP TO CYBERSECURITY FRAMEWORKS

CART can be mapped conceptually to the NIST Cybersecurity Framework 2.0 functions:

*1)* Govern;
*2)* Identify;
*3)* Protect;
*4)* Detect;
*5)* Respond;
*6)* Recover.

The framework particularly contributes to:

Detect
through cross-platform anomaly detection;
Respond
through exposure assessment and isolation recommendations; and
Recover
through recovery-point trust evaluation.

NIST CSF 2.0 provides a high-level taxonomy of cybersecurity outcomes rather than prescribing one technical implementation. Therefore, CART should be regarded as an implementation-oriented research framework rather than a replacement for a cybersecurity framework.

## XXXV. LIMITATIONS

The proposed framework has several limitations.
First, it does not guarantee ransomware prevention.
Second, machine-learning models may be vulnerable to adversarial manipulation.
Third, recovery trust cannot be determined purely from telemetry in every incident.
Fourth, data dependencies may be incomplete in highly heterogeneous environments.
Fifth, model performance depends on telemetry quality and appropriate baseline construction.
Finally, a laboratory evaluation cannot fully reproduce the operational complexity of a major financial institution.
These limitations should be explicitly reported rather than hidden.

## XXXVI. FUTURE RESEARCH

Future work should investigate:

*1)* federated learning across financial institutions without sharing raw telemetry;
*2)* graph neural networks for infrastructure dependency modeling;
*3)* continual learning under ransomware concept drift;
*4)* adversarial manipulation of infrastructure telemetry;
*5)* cryptographically verifiable recovery provenance;
*6)* autonomous but policy-constrained recovery;
*7)* multi-site sovereign-cloud architectures;
*8)* integration with Security Information and Event Management systems;
*9)* integration with Security Orchestration, Automation and Response systems;
*10)* formal verification of recovery decisions.

A particularly promising direction is combining the Recovery Trust Score with a probabilistic graphical model so that uncertainty is explicitly represented rather than hidden inside a deterministic score.

## XXXVII. CONCLUSION

This paper proposed CART, an AI-driven cross-platform anomaly-detection and recovery-trust framework for mission-critical banking data infrastructure.
Unlike conventional ransomware detection approaches that primarily classify activity as malicious or benign, CART treats cyber resilience as a continuous detection-to-recovery problem.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The framework combines telemetry from containers, virtualization, enterprise storage, object storage, backup, and replication systems. It introduces context-aware temporal anomaly detection, cross-platform behavioral correlation, concept-drift monitoring, explainable decision support, and a Recovery Trust Score for candidate recovery points.

The central research proposition is that ransomware resilience should not end when an attack is detected. A financial institution must also determine whether its available recovery states remain trustworthy.

The proposed experimental methodology is designed to test whether cross-platform behavioral correlation can improve early detection and whether recovery-point trust scoring can reduce unsafe recovery decisions compared with conventional recovery selection.

The empirical claims of the final publication must be based exclusively on the results of the proposed experiments. This is essential for scientific validity and publication integrity.

### *A. Declarations*

*1)* Funding: No external funding has been declared for the present manuscript. This statement should be updated if research funding is subsequently obtained.

*2)* Conflict of Interest: The authors should declare all financial, professional, commercial, or other relationships that could reasonably be perceived as influencing the research.

*3)* Data Availability: Public datasets used in the study will be identified by version, source, and license. Synthetic telemetry generation scripts and experimental code should be made available where organizational and security constraints permit. Confidential production banking telemetry will not be published.

*4)* Ethics Statement: This research does not require human subjects or personal data when conducted exclusively with public datasets and synthetic infrastructure telemetry. If future experiments use organizational telemetry or identifiable information, the appropriate institutional, legal, privacy, and information-security approvals must be obtained before data collection and publication.

*5)* Author Contributions: Author contributions should be completed using the CRediT taxonomy after all contributors have made their actual contributions. Authorship should be limited to individuals who make substantial contributions to conception, methodology, software, investigation, analysis, or interpretation.

*6)* Generative AI Disclosure: Any use of generative AI in manuscript preparation should be disclosed according to the target journal's current policy. Generative AI should not be listed as an author. All technical claims, citations, experimental results, and conclusions must be independently verified by the human authors.

## REFERENCES

[1] Souppaya, M., Barker, W., Fisher, W., & Kent, K. (2026). Ransomware Risk Management: A Cybersecurity Framework 2.0 Community Profile. NIST IR 8374 Rev. 1.

[2] McBride, T., Ekstrom, M., Lusty, L. N., Sexton, J., & Townsend, A. (2020). Data Integrity: Recovering from Ransomware and Other Destructive Events. NIST Special Publication 1800-11. DOI: 10.6028/NIST.SP.1800-11.

[3] Hirano, M., & Kobayashi, R. (2024). RanSMAP: Open Dataset of Ransomware Storage and Memory Access Patterns for Creating Deep Learning Based Ransomware Detectors. Computers & Security, 104202. DOI: 10.1016/j.cose.2024.104202.

[4] Onwuegbuche, F. C., Adelodun, S. O., Jurcut, A. D., & Pasquale, L. (2026). MLRan: A behavioural dataset for ransomware analysis and detection. Journal of Network and Computer Applications, 250, 104475. DOI: 10.1016/j.jnca.2026.104475.

[5] Fernando, D. W., & Komninos, N. (2024). FeSAD ransomware detection framework with machine learning using adaption to concept drift. Computers & Security, 137, 103629. DOI: 10.1016/j.cose.2023.103629.

[6] Tayouri, D., Sgan Cohen, O., Maimon, I., Mimran, D., Elovici, Y., & Shabtai, A. (2025). CORAL: Container Online Risk Assessment with Logical attack graphs. Computers & Security, 150, 104296. DOI: 10.1016/j.cose.2024.104296.

[7] [Containerized intrusion-detection study]. (2025). Enhancing intrusion detection in containerized services: Assessing machine learning models and an advanced representation for system call data. Computers & Security, 154, 104438. DOI: 10.1016/j.cose.2025.104438.

[8] [Adversarial ransomware study]. (2026). GUARD: Graph-based utility for adversarial ransomware detection using structural and behavioural characteristics. Array, 30, 100911.

[9] Kubernetes Documentation. Observability: Metrics, Logs and Audit Information. Kubernetes Project.

[10] [Software/container anomaly detection study]. (2025). Software anomaly detection technology based on deep learning. Procedia Computer Science, 259, 1123–1129.

[11] Tayouri, D., Sgan Cohen, O., Maimon, I., Mimran, D., Elovici, Y., & Shabtai, A. (2025). CORAL: Container Online Risk Assessment with Logical attack graphs. Computers & Security, 150, 104296.

[12] Red Hat. OpenShift Container Platform: Backup and Restore Documentation. Red Hat Documentation.

[13] Pascoe, C., Quinn, S., & Scarfone, K. (2024). The NIST Cybersecurity Framework (CSF) 2.0. NIST Cybersecurity White Paper CSWP 29. DOI: 10.6028/NIST.CSWP.29.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Appendix A. Proposed Feature Taxonomy

| Domain | Example features |
|---|---|
| OpenShift/Containers | pod events, restarts, CPU, memory, network, filesystem, audit activity |
| Virtualization | VM CPU, memory, disk I/O, snapshots, lifecycle events |
| Block/File Storage | IOPS, throughput, latency, queue depth, changed blocks |
| Object Storage | PUT, GET, DELETE, versions, object sizes, policy changes |
| Backup | job duration, size, failures, deletion, immutability |
| Replication | lag, changed data, throughput, synchronization |
| Recovery | timestamp, provenance, integrity, exposure, protection state |

Appendix B. Proposed Experimental Scenarios

| Scenario | Purpose |
|---|---|
| Normal workload | Baseline |
| Database backup | High-volume benign activity |
| ETL | False-positive testing |
| Storage migration | Behavioral variation |
| OpenShift deployment | Container-context testing |
| Object bulk upload | Object-storage baseline |
| Replication resynchronization | High-throughput benign activity |
| Fast ransomware | Early detection |
| Slow ransomware | Evasion testing |
| Multi-platform ransomware | Cross-platform correlation |
| Backup manipulation | Recovery trust |
| Snapshot exposure | Recovery-point assessment |
| Mixed benign + attack | Realistic noisy environment |
| Unseen ransomware family | Generalization |
| Temporal drift | Model longevity |

Appendix C. Proposed Main Research Figures
Figure 1: CART architecture.
Figure 2: Cross-platform telemetry pipeline.
Figure 3: Context-aware behavioral baseline.
Figure 4: Cross-platform anomaly correlation graph.
Figure 5: Recovery Trust Score architecture.
Figure 6: Example ransomware propagation timeline.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Figure 7: ROC/PR curves.
Figure 8: Detection-time comparison.
Figure 9: Recovery-point ranking.
Figure 10: Ablation-study results.

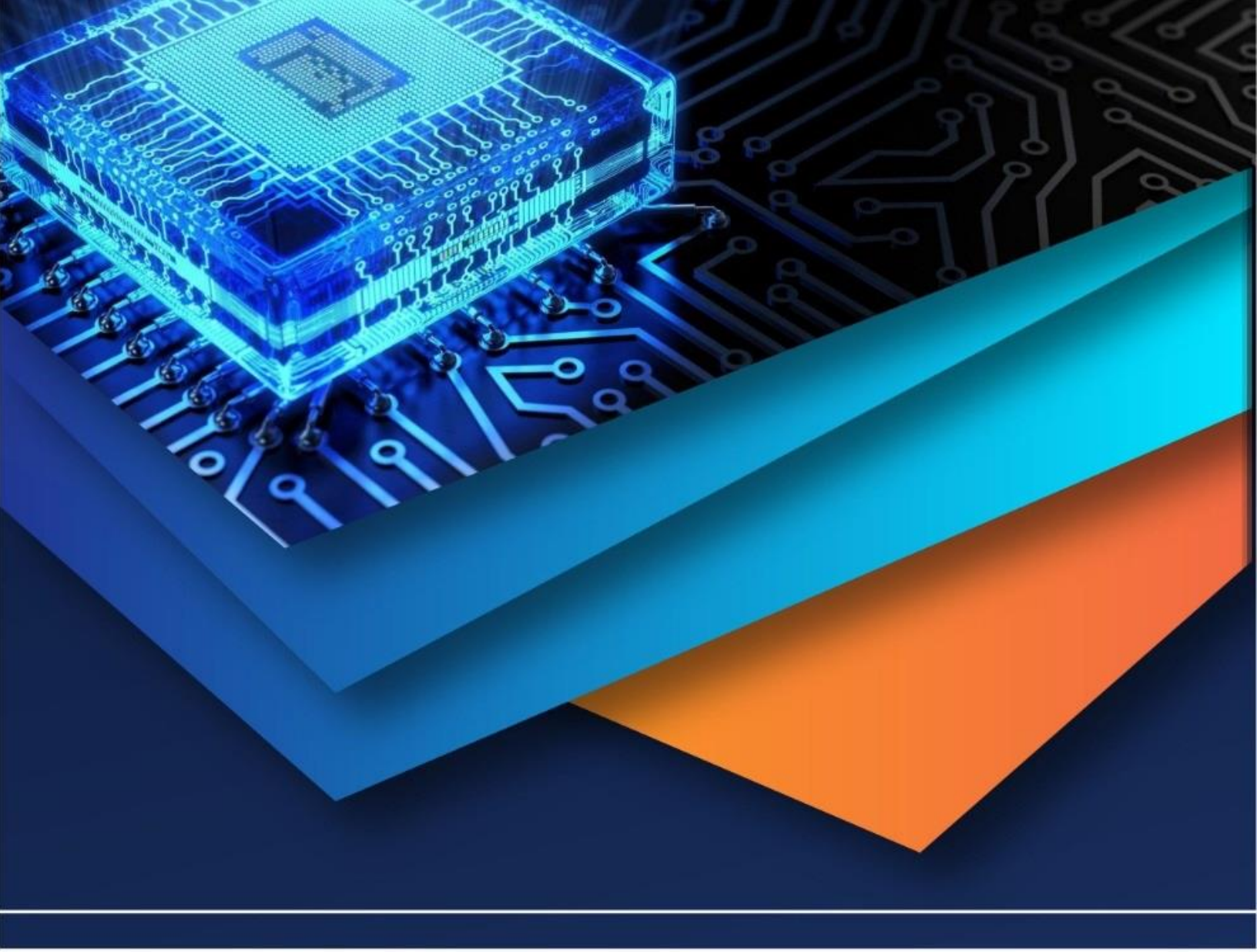
Appendix D. Core Research Claim

The paper should ultimately test the following proposition:

In heterogeneous mission-critical financial infrastructure, coordinated analysis of temporal telemetry across containers, virtualization, storage, object storage, backup, and replication can provide more informative ransomware detection and more reliable recovery-point selection than isolated platform-specific detection or latest-backup selection.

This proposition—not a claim of a predetermined accuracy percentage—is the central hypothesis of the research.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)