
Intelligent Ransomware Detection Using Behavioral Analytics and Hybrid Machine Learning Classification Techniques

Nazarova Nilufar¹, Sanyam Jain²

¹Department of Social Subjects, Tashkent State Transport University, Tashkent, Uzbekistan

nazarova_n@tstu.uz

0000-0002-2201-1216

²Researcher

sanyam.055898@tmu.ac.in

Abstract

Ransomware is a rapidly evolving cyber threat that exploits legitimate file-system operations, process execution, and network communication to encrypt or exfiltrate organizational information. This paper proposes a behavior-centric hybrid machine learning framework that combines gradient-boosted decision trees with temporal sequence modeling to identify ransomware activity from short windows of endpoint telemetry. The architecture integrates event collection, process-level correlation, feature engineering, temporal classification, probability calibration, and risk-aware alerting. Particular attention is given to preventing label leakage, evaluating unseen ransomware families, and controlling false alarms under realistic workload shifts. An explicitly illustrative numerical case study is provided to demonstrate the reporting protocol, including confusion matrices, detection metrics, latency, and ablation analysis; the values are not claimed to be obtained from an executed experiment. The proposed design offers a reproducible research blueprint for subsequent empirical validation with safely collected benign and ransomware traces.

Keywords: Ransomware detection; behavioral analytics; hybrid machine learning; gradient boosting; temporal neural networks; endpoint security; early warning; explainable AI.

DOI: 10.18091/ijsts.v9i3.08

1. Introduction

Ransomware is a class of malicious software designed to deny access to information or systems, commonly through encryption and an accompanying extortion demand. Contemporary campaigns may also steal data, interrupt recovery mechanisms, and pressure victims through disclosure threats. These behaviors complicate the conventional view of ransomware as merely a cryptographic program. Effective detection must therefore recognize an unfolding chain of actions rather than wait for a ransom note or a known executable signature. For enterprises, universities, hospitals, and

public agencies, the operational objective is to interrupt malicious activity early enough to reduce irreversible file modifications while preserving legitimate workloads.

Traditional signature-based antimalware engines match known byte patterns, hashes, rules, or previously observed indicators. They remain useful for recognized threats, but their performance can decline when malware authors repackage binaries, modify execution chains, or deploy new variants. Behavioral monitoring instead examines observable actions such as unusually rapid file rewrites, mass renaming, changes in file entropy, access to shadow-copy facilities, process ancestry, and bursts of suspicious API calls. A benign compression utility or backup client can also produce unusual file activity, so individual signals are insufficient. Detection becomes a contextual classification problem requiring joint analysis of process identity, event timing, operation frequency, and the surrounding system state.

An important distinction is the difference between detecting ransomware after substantial encryption has occurred and detecting the attack at an actionable early stage. A classifier evaluated on complete execution traces may appear accurate while being operationally too slow. A practical system must process short, overlapping windows and measure the time from first malicious activity to alert. The percentage of files modified before intervention is another meaningful outcome. Both outcomes depend on telemetry collection delays, feature computation, inference time, alert thresholds, and endpoint response policy. Detection accuracy alone does not adequately represent protective utility.

Behavioral data arise from several sources. Endpoint sensors can record file create, open, write, rename, and delete events; process creation and termination; parent-child process relationships; registry modifications; command-line activity; and selected network connections. Kernel-level or operating-system tracing can provide detailed visibility but may create computational and privacy costs. Enterprise deployment therefore demands selective instrumentation, bounded event buffering, and aggregation into interpretable process-window features. The collection layer should minimize sensitive content capture by storing metadata rather than full document contents whenever feasible.

Machine learning offers mechanisms for learning patterns from high-dimensional behavioral observations. Linear models provide transparent baselines and efficient inference but may struggle with nonlinear interactions. Random forests and gradient-boosted trees learn complex feature relationships from aggregated counts, rates, and statistical summaries. Recurrent networks, temporal convolutional networks, and attention-based sequence models can learn event order and temporal dependencies. Their relative utility depends on dataset size, sequence quality, available compute, and the degree of behavior variation between ransomware families. A hybrid design is motivated by the complementary strengths of structured-feature models and sequential learners.

The proposed framework uses two coordinated detection branches. A gradient-boosted tree model receives aggregated behavioral features, including write intensity, rename rate, extension diversity, process lineage attributes, and entropy change. A gated recurrent unit (GRU) receives a fixed-length sequence of event embeddings and time-gap features. The outputs are calibrated and fused into a single risk score. This combination allows the system to recognize both unusual aggregate intensity

and characteristic ordering of events, while retaining a fast screening path for low-risk activity. The fusion threshold is selected on validation data with explicit attention to false-positive constraints.

Class imbalance is a central challenge. In ordinary enterprise activity, benign processes vastly outnumber ransomware executions. A model that predicts benign behavior nearly all the time can exhibit high overall accuracy while missing rare attacks. Evaluation must consequently emphasize recall, precision, F1-score, area under the precision-recall curve, and false-positive rate at fixed operating points. It should also report alert counts per endpoint-day, because even a seemingly small false-positive rate can overwhelm analysts when millions of events are screened. Probability calibration and threshold selection are therefore operationally relevant, not optional refinements.

Another challenge is generalization. Randomly splitting windows from the same executable across training and testing can produce optimistic estimates because adjacent windows share event patterns and execution context. Stronger validation separates ransomware families, executable hashes, hosts, and time periods. This paper recommends a primary family-disjoint test set and a secondary chronological holdout. Benign samples should include demanding workloads such as software installation, backup, bulk document conversion, encryption utilities, and database maintenance. These hard negatives are needed to distinguish malicious intent from ordinary high-volume activity.

Interpretability also matters. Security teams require evidence that an alert reflects a meaningful behavior pattern, rather than an incidental artifact such as a file path or host identifier. Feature attribution for the tree branch, event saliency for the sequence branch, and a concise timeline of contributing operations can support analyst review. Explanations must not be mistaken for causal proof, and the system should preserve raw supporting telemetry for authorized forensic investigation. Privacy, auditability, and governance become especially important when the detector monitors employee endpoints and institutional systems.

The contributions of this paper are fourfold: a process-centered behavioral representation; a calibrated hybrid tree–sequence classifier; an evaluation protocol designed to reduce leakage and quantify early detection; and an illustrative reporting example with numerical tables and graphics. The numerical case study is deliberately marked as hypothetical because this manuscript does not claim access to executed sandbox experiments or measured telemetry. The resulting paper should be understood as a detailed proposed-method study and validation plan, rather than a completed empirical demonstration of superiority. Future implementation and independently reproducible experiments are necessary before the system can be recommended for production deployment.

2. Related Work

Early ransomware detection studies investigated file-system monitoring and the use of behavioral heuristics to identify high-frequency file modifications and cryptographic activity. Such methods are attractive because they observe consequences of execution without requiring access to malware source code. However, threshold-only rules can misclassify legitimate backup, synchronization, and bulk-processing applications. A further limitation is that a single rate threshold may be too sensitive

for one endpoint and insufficiently sensitive for another. These observations motivate models that combine multiple correlated signals and account for context.

Process behavior analysis offers a richer basis for classification. Studies examining process events have shown the feasibility of distinguishing ransomware from benign software using runtime activity rather than static identifiers. For example, the 2020 work “Detecting Ransomware Using Process Behavior Analysis” considered ransomware, benign applications, and other malware samples and highlighted the promise of process-centric analysis [1]. Nevertheless, results obtained from relatively small collections can be vulnerable to family bias, sample-selection effects, and uncertainty about performance on novel threats. Reproducible train-test separation is essential.

Static analysis remains a complementary technique. Portable executable headers, imported libraries, strings, and byte-level characteristics can identify suspicious binaries before execution. These signals are inexpensive to collect but may be obscured by packing, obfuscation, or dynamically loaded code. Dynamic analysis observes execution behavior and is more directly aligned with harmful file operations, yet sandbox traces can be incomplete when malware delays activation or detects instrumentation. A hybrid endpoint system may use static information as auxiliary context without making the dynamic classifier dependent on it.

Machine learning approaches based on decision trees, random forests, support vector machines, and gradient boosting have been explored for malware and ransomware detection. Tree ensembles are particularly useful for tabular telemetry because they model nonlinear interactions and tolerate heterogeneous feature scales. Their weaknesses include sensitivity to training-distribution shifts and inability to represent long event sequences unless temporal summaries are engineered. Strong validation should therefore compare boosted trees against simple baselines under the same family-disjoint partition, rather than treating a high score on a convenient random split as definitive.

Deep learning methods attempt to represent sequential behavior through recurrent neural networks, temporal convolution, and attention mechanisms. Sequence models can distinguish a burst of ordinary file activity from a progression that includes discovery, repeated file opens, write bursts, renaming, and recovery interference. However, a sequence model can also overfit execution length, instrumentation artifacts, or benign application identities. Its performance depends on event-token design, truncation policy, temporal alignment, and the quality of labels attached to short windows. Latency and memory consumption must be reported alongside classification metrics.

A comprehensive 2024 review of machine-learning-based ransomware detection emphasized the gap between promising academic classifications and operational early-warning designs [2]. This distinction is important because offline classification of completed traces does not demonstrate the ability to stop an attack in progress. Evaluation must reflect how soon a detector makes a decision, the proportion of affected files at that moment, and the overhead introduced by monitoring. These measures are particularly significant on shared workstations, file servers, and resource-constrained endpoints.

Alzahrani and colleagues surveyed ransomware detection methods published through 2025 and classified machine-learning and non-machine-learning techniques while examining datasets and

practical limitations [3]. Their synthesis underscores the breadth of operating-system and analysis settings. A common difficulty in comparing studies is that dataset composition, family coverage, class prevalence, and reported metrics vary substantially. Accuracy reported for one dataset cannot be interpreted as directly superior to accuracy reported for another. The present proposal accordingly emphasizes matched baselines and identical evaluation partitions.

Recent surveys have drawn attention to behavioral, pre-encryption, and hybrid detection, as well as adversarial robustness and explainability [4]. A system that relies on a few prominent signals may be evaded by slowing encryption, spreading operations across processes, or interleaving benign activity. Defenders must test low-and-slow scenarios, delayed execution, and multi-process coordination in authorized, isolated settings. The proposed framework uses short temporal windows and process ancestry correlations, but it does not claim immunity to adversarial adaptation. Robustness is an empirical question that requires explicit challenge sets.

Contemporary work has also explored pairing fast boosted classifiers with sequential models. A 2025 behavioral framework combining XGBoost and LSTM with threat intelligence illustrates the practical appeal of multi-stage detection [5]. Such designs can reduce computational cost by reserving expensive inference for ambiguous observations. Yet reported performance can be difficult to reproduce without accessible code, exact feature definitions, sample identifiers, and a detailed separation protocol. Our proposed architecture retains the two-branch principle while specifying probability calibration, validation-only threshold tuning, and deployment-focused alert metrics.

RansomTrack, described in a 2026 preprint, combines static and dynamic behavioral evidence and reports ensemble-based detection with interpretability [6]. It illustrates growing interest in transparent and deployable hybrid systems. Preprint findings should be distinguished from independently replicated evidence, and dataset novelty alone does not ensure representativeness of enterprise workloads. The proposal in this paper prioritizes process-window labeling, family-disjoint evaluation, and explicit measurement of detection timing to address several persistent comparability challenges.

The literature also reveals important data-governance constraints. Ransomware samples are hazardous and should be executed only within controlled research environments with containment and authorization. Endpoint telemetry can expose filenames, user identities, and organizational routines; consequently, datasets should be de-identified and retained according to a documented policy. Cross-organization learning is promising but introduces distribution shift and privacy challenges. Any claim that a model generalizes across enterprises should be based on host- or organization-disjoint evaluation, not on a random mixture of records from the same source.

Taken together, prior work supports behavioral machine learning as a useful direction but does not eliminate fundamental questions about early warning, false alarms, reproducibility, and deployment cost. The specific research gap addressed here is the lack of a consistently specified, latency-aware hybrid evaluation protocol connecting aggregated process statistics with event-order modeling and risk-calibrated decisions. The proposed contribution is methodological: it defines the data pipeline, equations, fusion mechanism, and evaluation outputs required to test the hypothesis that hybrid

modeling improves detection under matched conditions. No empirical superiority is asserted without a real experiment.

3. Proposed Methodology

3.1 Threat model and telemetry acquisition

The defender monitors authorized Windows or Linux endpoints. The adversary may launch ransomware that rapidly or gradually modifies accessible files, potentially through multiple related processes. The system observes metadata-level file, process, registry, and network events; it does not assume access to decrypted payloads. Events are keyed by host, process, process ancestry, and timestamp. A five-second sliding window with a two-second stride is proposed as an initial configuration to be tuned experimentally. Collection failures, clock skew, and missing events are recorded as quality indicators.

3.2 Feature extraction and preprocessing

For each process-window, features include write operations per second, rename-to-write ratio, number of distinct file extensions, median bytes written, mean entropy delta where safely measurable, number of unique directories touched, suspicious recovery-tool access, parent-process risk, and network connection counts. Numerical values are clipped using training-set bounds and standardized using training-only statistics. Categorical event types are embedded for the temporal branch. Hostnames, user IDs, and absolute paths are excluded from predictive features to reduce leakage and privacy risk.

3.3 Hybrid classification and equations

$$r_write = N_write / (\Delta t + \varepsilon) \quad (1)$$

Here N_write is the count of write operations within a window, Δt is window duration in seconds, and ε is a small positive constant. This rate measures activity intensity independent of window length.

$$H(X) = - \sum_i p_i \log_2(p_i) \quad (2)$$

Shannon entropy H is calculated from byte-frequency probabilities p_i in an authorized sample of file content; entropy change is the difference between post-write and pre-write estimates. High entropy alone is not evidence of ransomware.

$$p_tree = \sigma(F_GB(x)) \quad (3)$$

$F_GB(x)$ is the gradient-boosting logit computed from tabular feature vector x ; $\sigma(z)=1/(1+e^{(-z)})$ maps the score into a probability-like value before calibration.

$$h_t = GRU(e_t, h_{t-1}), \quad p_seq = \sigma(w^T h_T + b) \quad (4)$$

The GRU updates a hidden state using event embedding e_t and the previous hidden state. The last state h_T summarizes temporal behavior, and a sigmoid classifier produces the sequence score.

$$p_fuse = \alpha \cdot Cal(p_tree) + (1-\alpha) \cdot Cal(p_seq) \quad (5)$$

Cal denotes validation-fitted probability calibration; α lies between zero and one and is selected on validation data. Fusion balances aggregated and sequential evidence.

$$\hat{y} = 1[p_{\text{fuse}} \geq \tau] \quad (6)$$

The decision threshold τ is tuned on the validation partition to satisfy a target false-positive rate; the test set is never used for threshold selection.

$$L = -(1/N)\sum_i [w_1 y_i \log(p_i) + w_0 (1 - y_i) \log(1 - p_i)] \quad (7)$$

Weighted binary cross-entropy assigns class weights w_1 and w_0 to address imbalance. Training weights must be estimated using training data only.

3.4 Detection workflow

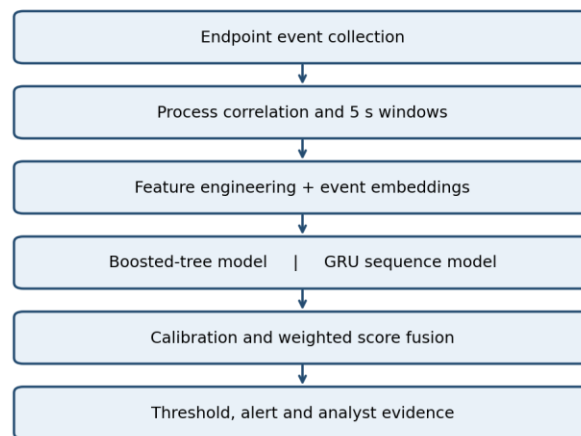


Figure 1. Proposed behavior-driven hybrid ransomware detection pipeline.

3.5 Training and evaluation protocol

Samples should be collected only in an isolated, authorized sandbox and supplemented with consented benign enterprise traces. Split by ransomware family and executable identity before creating overlapping windows. Reserve training, validation, and held-out test partitions; a separate chronological test assesses drift. Train logistic regression, random forest, boosted trees, GRU, and the fused model under matched inputs. Tune all hyperparameters on validation data. Report five random seeds, confidence intervals, class prevalence, PR-AUC, ROC-AUC, FPR, detection delay, throughput, memory overhead, and the number of modified files at first alert. Investigate benign hard negatives and missing telemetry. No live malware was executed for this manuscript.

4. Illustrative Results and Discussion

The numerical values in Tables 1–3 and Figures 2–3 are a constructed example showing how an eventual experiment should be reported. They are not measurements from a real dataset, published benchmark, or implemented prototype. The illustrative test cohort contains 2,000 windows (1,000

ransomware-positive and 1,000 benign), an artificially balanced prevalence chosen only for transparent arithmetic; operational prevalence would normally be substantially lower.

4.1 Numerical model comparison

Table 1. Illustrative (hypothetical) model comparison; latency is per-window inference time and excludes event collection

Classifier	Accuracy %	Precision %	Recall %	F1 %	Latency ms
Logistic regression	92.0	91.7	92.4	92.0	5.5
Random forest	95.0	94.6	95.5	95.0	8.2
Gradient boosting	96.2	95.8	96.8	96.3	7.4
GRU sequence model	96.7	96.3	97.2	96.7	13.8
Proposed hybrid	97.8	97.5	98.1	97.8	16.5

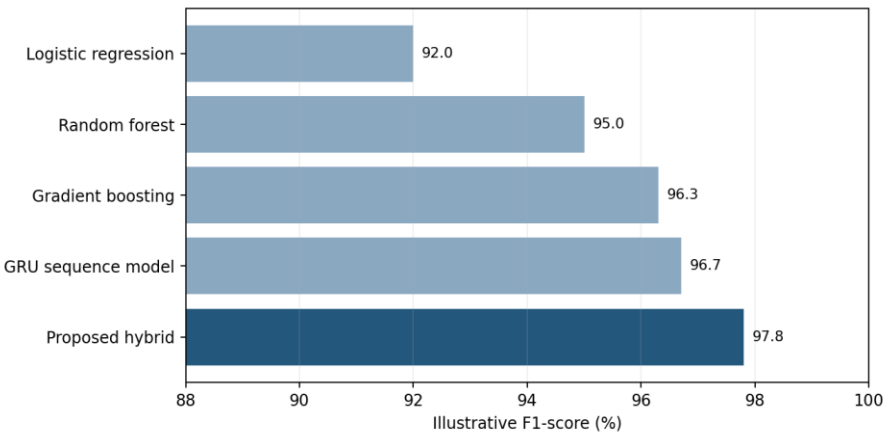


Figure 2. Hypothetical F1-score comparison, based on the constructed values in Table 1.

In this illustrative comparison, the fused classifier yields the highest F1-score, while logistic regression has the lowest inference latency. The apparent improvement from 96.7% for the GRU to 97.8% for fusion is a design illustration, not an experimentally established gain. A real study should assess whether the difference persists across independent families and whether the additional computation is acceptable.

4.2 Confusion matrix and derived measures

Table 2. Illustrative confusion matrix for the proposed hybrid classifier (2,000 windows)

Actual / predicted	Ransomware	Benign
--------------------	------------	--------

Ransomware	981 (TP)	19 (FN)
Benign	25 (FP)	975 (TN)

From Table 2, $\text{accuracy} = (981 + 975) / 2000 = 97.8\%$; $\text{precision} = 981 / (981 + 25) = 97.51\%$; $\text{recall} = 981 / (981 + 19) = 98.10\%$; $\text{F1} = 2 \times 981 / (2 \times 981 + 25 + 19) = 97.80\%$; $\text{specificity} = 975 / (975 + 25) = 97.50\%$; and $\text{FPR} = 25 / (25 + 975) = 2.50\%$. Table 1 uses the confusion-matrix-consistent rounded precision of 97.5%. An FPR of 2.5% would be far too high for many high-volume production environments, underscoring the need for threshold tuning at much lower operating FPRs.

4.3 Ablation and early-warning trade-offs

Table 3. Hypothetical ablation values illustrating the contribution of score calibration

Configuration	F1-score %	Latency ms
Tree branch only	96.3	7.4
Sequence branch only	96.7	13.8
Uncalibrated fusion	97.1	15.9
Calibrated fusion	97.8	16.5

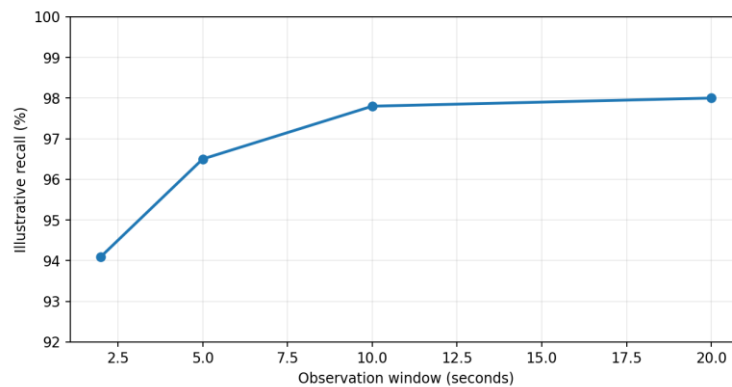


Figure 3. Hypothetical recall versus observation window duration; larger windows may delay intervention.

The hypothetical ablation suggests that combining calibrated branches may increase classification performance, but the experiment must verify whether the gain exceeds statistical uncertainty. A longer observation window can improve recognition by exposing more behavior while delaying alerts and allowing more file modifications. Therefore, deployment should optimize a multi-

objective criterion that balances recall, false-positive burden, inference latency, and time-to-detection. The illustrative results do not support claims about zero-day performance, ransomware prevention, or generalization to real enterprise networks.

5. Limitations and Reproducibility

The primary limitation is the absence of executed experiments; all numerical comparisons are demonstrative. Further limitations include the need for diverse, legally obtained ransomware traces, incomplete visibility on encrypted or cloud-hosted workloads, adversarial evasion, endpoint overhead, and privacy constraints. A reproducible study should release feature definitions, anonymized split identifiers, training configurations, seeds, threshold selection code, and evaluation scripts wherever legally possible. Confidence intervals should be computed over independent runs or independent ransomware families rather than over highly correlated windows.

6. Conclusion

This paper developed a detailed methodology for intelligent ransomware detection through behavioral analytics and hybrid machine learning. The proposed architecture fuses a gradient-boosted classifier operating on aggregated process features with a GRU model operating on event sequences. Calibration, family-disjoint testing, short-window inference, and false-positive-aware threshold selection are integral to the design. The constructed numerical tables and graphs illustrate how model comparisons, confusion matrices, and ablation studies should be reported, but they do not constitute empirical evidence of performance. The central research finding at this stage is methodological: reliable ransomware detection research must evaluate early-warning utility and cross-family generalization rather than classification accuracy alone. Future work will implement the pipeline, evaluate it on authorized multi-family traces and realistic benign workloads, quantify the proportion of files modified before alert, and test drift, adversarial resilience, and operational overhead.

References

1. Mahajan, R. A., Dey, R., Khan, M., Kanabar, K., & Gupta, S. K. (2025). Real-time vehicular-to-vehicular (V2V) communication and applications for pedestrian safety in AI-optimized VANETs for autonomous vehicles. In H. Maryam, M. M. Malik, I. U. Khan, & S. K. Gupta (Eds.), *AI-driven transportation systems: Real-time applications and related technologies* (pp. 151–167). Springer. https://doi.org/10.1007/978-3-031-98349-8_8
2. Bhaskara Raju Rallabandi. (2024). 6G Cross-Layer Intent Engine: Automated Translation from Business Requirements to Network Policies and Real-Time Scheduling Controls. *International Journal of Data Science and IoT Management System*, 3(1), 44–48. <https://doi.org/10.64751/ijdim.2024.v3.n1.pp44-48>
3. Aggarwal, A. (2024). Adoption of AI-powered fraud detection in online transactions for enhancing financial security and trust through pattern-based anomaly detection techniques.

-
- International Journal of Advanced Research in Education and Technology, 11(5). <https://doi.org/10.15680/IJARETY.2024.1105043>
4. Devulapalli, N. (2020). The evolution of database security from centralized architectures to distributed cloud-native systems: A comparative and historical perspective. *Turkish Online Journal of Qualitative Inquiry*, 11(4), 2954–2963. <https://doi.org/10.69980/j1064z97>
 5. Rao, S. Madhusudhana, and Arpit Jain. "Advances in Malware Analysis and Detection in Cloud Computing Environments: A Review." *International Journal of Safety & Security Engineering* 14, no. 1 (2024).
 6. Devulapalli, N. (2018). Database security for financial systems: Protecting customer data, ensuring PCI-DSS compliance, and safeguarding against fraud and insider attacks. *International Journal of Advanced Research in Education and Technology*, 5(5), 838–844. <https://doi.org/10.15680/IJARETY.2018.0505018>
 7. Ahuja, R. (2020). Adoption of secure API development practices in microservice architectures for preventing unauthorized access and data leakage through OAuth2 and input throttling. *Journal of Critical Reviews*, 7(9), 4063–4069. <https://doi.org/10.69980/jcr.v7:i9.13404>
 8. Mahadasu, S. S. (2022). Cross-domain service assurance for private 5G networks using streaming telemetry and dependency-aware event correlation. *International Journal of Data Science and IoT Management System*, 1(3), 51–56. <https://doi.org/10.64751/ijdim.2022.v1.n3.1290>
 9. Kumar, S., Sharma, K., Kumar, P. A., Jain, A., Bhagat, S. K., & Singh, P. (2024, September). An improved particle swarm approach for energy-aware location-aided routing in mobile ad-hoc network. In *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)* (Vol. 7, pp. 1119–1124). IEEE.
 10. Bhaskara Raju Rallabandi. (2023). AGENTIC-AI xAPPS: O-RAN MULTI-AGENT CONTROL FOR ENTERPRISE POLICY, SCHEDULING & SPECTRUM. *International Journal of Data Science and IoT Management System*, 2(4), 54–58. <https://doi.org/10.64751/ijdim.2023.v2.n4.pp54-58>
 11. Aggarwal, A. (2025). Artificial intelligence in criminal justice systems: An examination of predictive policing, automated surveillance, and ethical concerns in legal decision-making. *International Journal of Multidisciplinary Research in Science, Engineering and Technology*, 8(1). <https://doi.org/10.15680/IJMRSET.2025.0801070>
 12. Anchala, S. (2024). Deployment of zero trust security models in hybrid cloud infrastructures for minimizing lateral movement and strengthening access control through continuous verification mechanisms. *International Journal of Innovative Research in Computer and Communication Engineering*, 12(7), 9993–10002. <https://doi.org/10.15680/IJRCCE.2024.1207114>
 13. Aggarwal, A. (2021). Utilization of continuous vulnerability scanning in DevSecOps pipelines for preemptive threat detection and secure code deployment through automated security checks. *NeuroQuantology*, 19(1), 427–442. <https://doi.org/10.48047/nq.2021.19.1.NQ21055>
 14. Gupta, Namit, Kunwar Singh Vaisla, Arpit Jain, Adesh Kumar, and Rajeev Kumar. "Performance Analysis of AODV Routing for Wireless Sensor Network in FPGA Hardware." *Computer Systems Science & Engineering* 40, no. 3 (2022).

15. Mahadasu, S. S. (2024). AI-native wireless digital twins for precision agriculture: Edge intelligence and predictive farm operations in 6G ecosystems. *International Journal of Data Science and IoT Management System*, 3(1), 65–72. <https://doi.org/10.64751/ijdim.2024.v3.n1.1287>
16. Harshitha, G., Kumar, S., Rani, S., & Jain, A. (2021, November). Cotton disease detection based on deep learning techniques. In *4th Smart Cities Symposium (SCS 2021)* (Vol. 2021, pp. 496–501). IET.
17. Mahajan, R. A., Dey, R., Khan, M., Mohd Su'ud, M., Alam, M. M., & Jadhav, P. (2025). Enhancing personalization in IoT-based health monitoring via generative AI and transfer learning. *Egyptian Informatics Journal*, 32, Article 100788. <https://doi.org/10.1016/j.eij.2025.100788>
18. Mahadasu, S. S. (2024). Closed-loop sustainability orchestration for energy-aware RAN, edge, and data-center workloads in 5G-Advanced networks. *International Journal of Engineering Research and Science & Technology*, 20(2), 1392–1398. <https://doi.org/10.62643/ijerst.2024.v20.n2.4407>
19. Bhaskara Raju Rallabandi. (2024). DESIGNING NTN–TERRESTRIAL 6G EXPERIMENTS FOR AGRICULTURE AND SMART GRIDS. *American Journal of Management and IOT Medical Computing*, 3(3), 27–32. <https://doi.org/10.64751/ajmimc.2024.v3.n3.pp27-32>
20. Jain, Arpit, Adesh Kumar, and Sanjeev Sharma. "Comparative design and analysis of mesh, torus and ring NoC." *Procedia Computer Science* 48 (2015): 330–337.
21. Dwivedi, D. (2018). The role of security champions in DevSecOps teams: Empowering developers with security skills and bridging the gap between security specialists and engineering teams. *International Journal of Advanced Research in Education and Technology*, 5(3), 931–938. <https://doi.org/10.15680/IJARETY.2018.0503016>
22. Ahuja, R. (2019). A comparative study of traditional DevOps and DevSecOps: Examining the cultural, technical, and organizational shifts required to embed security into software engineering practices. *International Journal of Innovative Research in Science, Engineering and Technology*, 8(8), 8561–8569. <https://doi.org/10.15680/IJRSET.2019.0808096>
23. Devulapalli, N. (2023). Database security and big data analytics: Protecting large-scale data warehouses, ensuring query privacy, and mitigating emerging threats in real-time analytics systems. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(7), 622–629. <https://www.ijritcc.org/index.php/ijritcc/article/view/11939>
24. Jain, A. (2024, December). Energy-Efficient Signal Processing Architectures for IoT Networks Design. In *2024 International Conference on Emerging Technologies and Innovation for Sustainability (EmergIN)* (pp. 362–367). IEEE.
25. Anchala, S. (2020). Adoption of secure access service edge (SASE) in distributed enterprises for ensuring cloud application protection and network optimization through unified security frameworks. *Turkish Online Journal of Qualitative Inquiry*, 11(4), 2964–2973. <https://doi.org/10.69980/60m26411>
26. Ahuja, R. (2020). Database security in smart cities: Protecting interconnected infrastructure data, ensuring citizen privacy, and managing large-scale IoT data streams. *International Journal of*

- Multidisciplinary and Scientific Emerging Research, 8(1), 410–418. <https://doi.org/10.15662/IJMSERH.2020.0801047>
27. Jain, Arpit, Nageswara Rao Moparthi, A. Swathi, Yogesh Kumar Sharma, Nitin Mittal, Ahmed Alhussen, Zamil S. Alzamil, and MohdAnul Haq. "Deep Learning-Based Mask Identification System Using ResNet Transfer Learning Architecture." *Computer Systems Science & Engineering* 48, no. 2 (2024): 341.
28. Khan, S. (2022). The use of artificial intelligence in ESG (environmental, social, and governance) investing: A study on AI models for sustainability analytics in asset and wealth management. *International Journal of Multidisciplinary and Scientific Emerging Research*, 10(1), 431–439. <https://doi.org/10.15662/IJMSERH.2022.1001046>
29. Mahadasu, S. S., & Rallabandi, B. R. (2023). Federated learning for privacy-preserving RAN optimization across multi-site enterprise 5G networks. *International Journal of Engineering Science and Advanced Technology*, 23(4), 53–57. <https://doi.org/10.64771/ijesat.2023.v23.i04.pp3847>
30. Aggarwal, A. (2025). Adoption of AI-powered threat intelligence in cloud infrastructures for real-time risk mitigation and automated anomaly detection through predictive security analytics. *Journal of Digital Security and Forensics*, 2(1), 139–150. <https://doi.org/10.29121/digisecforensics.v2.i1.2025.87>
31. Bhaskara Raju Rallabandi. (2023). CLOSED-LOOP DIGITAL-TWIN MANAGEMENT FOR HYBRID 5G-NTN NETWORKS. *American Journal of Management and IOT Medical Computing*, 2(3), 30–35. <https://doi.org/10.64751/ajmimc.2023.v2.n3.pp30-35>
32. Kumar, Ashok, and Arpit Jain. "Image smog restoration using oblique gradient profile prior and energy minimization." *Frontiers of Computer Science* 15, no. 6 (2021): 156706.
33. Khan, S. (2018). The role of zero-trust models in database security: Eliminating implicit trust and enforcing continuous verification in enterprise data access systems. *International Journal of Research in Electronics and Computer Engineering*, 6(1), 1622–1628.
34. Anchala, S. (2020). Cloud security challenges in supply chain systems: Addressing vendor risks, data integrity, and third-party service provider vulnerabilities. *Journal of Critical Reviews*, 7(2). <https://doi.org/10.69980/jcr.v7:i2.13401>
35. Ahuja, R. (2023). The role of artificial intelligence and machine learning in DevSecOps: Leveraging predictive analytics, automated threat detection, and anomaly identification for secure software delivery pipelines. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(2), 381–389. <https://www.ijritcc.org/index.php/ijritcc/article/view/11946>
36. Jain, Jaishree, Arpit Jain, Saurabh Kumar Srivastava, Chaman Verma, Maria Simona Raboaca, and Zoltán Illés. "Improved security of e-healthcare images using hybridized robust zero-watermarking and hyper-chaotic system along with RSA." *Mathematics* 10, no. 7 (2022): 1071.
37. Anchala, S. (2019). The evolution of DevSecOps in software supply chain security: Addressing dependencies, open-source vulnerabilities, and the prevention of software supply chain attacks. *International Journal of Multidisciplinary and Scientific Emerging Research*, 7(1), 415–422. <https://doi.org/10.15662/IJMSERH.2019.0701047>

38. Kumar, S., Jain, A., & Moparthi, N. R. (2022). Enhanced method of object tracing using extended Kalman filter via binary search algorithm.
39. Bhaskara Raju Rallabandi. (2022). SYNC-ADAPTIVE RESOURCE SCHEDULING IN PRIVATE 5G FOR TIME-CRITICAL APPLICATIONS. *International Journal of Data Science and IoT Management System*, 1(2), 40–45. <https://doi.org/10.64751/ijdim.2022.v1.n2.pp40-45>
40. Khan, S. (2019). Sales force security compliance: An in-depth study of GDPR, HIPAA, and PCI-DSS enforcement in cloud-based CRM systems and their implications for global enterprises. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 8(9), 2211–2219. <https://doi.org/10.15662/IJAREEIE.2019.0809010>