International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-15 Issue-5, April 2026

Behaviour-Aware Hybrid Deep Networks for Detecting Zero-Day and Ransomware Threats

Madhan Mohan Reddy Chinthala, Harish Apuri, Kishore Bitra

Abstract: Purpose: ***The rapid escalation of ransomware and zero-day malware attacks poses a significant challenge to conventional signature-based detection systems, which cannot generalise to previously unseen threats. This study aims to develop a robust, scalable, and behaviour-aware malware-detection framework capable of accurately identifying ransomware and zero-day attacks across heterogeneous computing environments.*** ***Design/methodology/approach: A novel multi-stage hybrid detection pipeline is proposed that integrates advanced feature selection, deep sequential learning, attention mechanisms, and ensemble classification. Initially, irrelevant and redundant features are eliminated using correlation thresholding, Chi-square analysis, mutual information, and variance-based ranking. To capture latent behavioral patterns, a hybrid Gated Recurrent Unit-Temporal Convolutional Network (GRU-TCN) architecture is employed to model long- and short-term temporal dependencies. These representations are further refined using squeeze-and-excitation attention-enhanced TCN blocks. Finally, an XG-Fusion framework that combines GRU encoding, dilated residual TCNs, attention-based feature fusion, and focal loss optimisation is introduced to address class imbalance, with XGBoost serving as a meta-classifier for final decision-making. Findings: Experimental evaluations conducted on multiple benchmark datasets demonstrate that the proposed framework consistently outperforms traditional machine learning and baseline deep learning models. Superior performance is achieved in terms of accuracy, precision, recall, F1 Score, and ROC AUC. The hierarchical and attention-driven architecture effectively abstracts malicious behavioral patterns and enhances generalization to previously unseen malware variants. Originality: This work introduces a novel multi-stage hybrid deep learning architecture that synergistically combines sequential behavioural modelling, attention-enhanced feature learning, and ensemble-based classification. The proposed approach offers a forward-looking and reliable solution for proactive detection of ransomware and zero-day malware threats.***

Keywords: Zero-Day Attack Detection, Ransomware Analysis, Hybrid Deep Learning (CNN-LSTM, VAE), Feature Selection & Behavioural Embedding, t-SNE Visualisation & Entropy-Based Analysis

Nomenclature:
GRU-TCN: Gated Recurrent Unit-Temporal Convolutional Network
SVD: Singular Value Decomposition
MI: Mutual Information

Manuscript received on 28 March 2026 | First Revised Manuscript received on 01 April 2026 | Second Revised Manuscript received on 07 April 2026 | Manuscript Accepted on 15 April 2026 | Manuscript published on 30 April 2026.
*Correspondence Author(s)
Madhan Mohan Reddy Chinthala*, Department of IT, Franklin Info Tech, Rochester (NY), United States of America (USA). Email ID: madhanreddychinthala@gmail.com, ORCID ID: 0009-0004-6236-7200
Harish Apuri, Department of IT, Induct Inc, Charlotte, NC, United States of America (USA). Email ID: hapuri1029@gmail.com, ORCID ID: 0009-0007-1534-9984
Kishore Bitra, Department of IT, Information and Technology, Baltimore (MD), United States of America (USA). Email ID: Kishore.bitra@baltimorecity.gov, ORCID ID: 0009-0007-1417-2742

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open-access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

I. INTRODUCTION

Cloud computing has brought about a paradigm shift in the design and management of IT resources, offering scalable, flexible, and cost-effective on-demand services for diverse applications, ranging from personal data storage to operations critical to the enterprise [3]. Security has been a critical issue as organizations move sensitive workloads and data to cloud set-ups in large numbers. Such a dependency on cloud infrastructure has also increased the attack surface, exposing systems to sophisticated cyber threats. Of these threats, zero-day attacks are particularly problematic because they exploit hitherto unknown software vulnerabilities that lack a set signature or patch [1]. Conventional security control systems, including signature-based intrusion detection systems, are largely ineffective against these attacks [2], [4]. In the cloud, shared resources and scalable deployments amplify the effects of zero-day attacks. Hence, it is of paramount importance that smart, dynamic security measures capable of identifying unusual activity and new threats that do not match the currently defined attack patterns are deployed to ensure the integrity and stability of cloud computing environments [5], [6].

II. LITERATURE REVIEW

Machine learning and deep learning are increasingly being used in cloud security to address the limitations of signature-based systems against unknown and zero-day threats [7], [8]. The hybrid CNN-RNN networks utilise spatial and temporal features, and cloud-specific feature engineering and attention control enhance robustness and reduce false alarms [9]. Logs, API calls, registry usage, and file access are useful behavioural indicators that can detect ransomware [10] - [18], and can be complemented with network-based anomaly detection [19] - [21]. Latent representations and feature selection are useful for maximising anomaly detection [22]-[27], whereas sequential behaviours are learned by temporal models such as LSTMs and temporal CNNs [28]-[30]. The threats persist in the invisible threat management and the inspiration for hybrid multi-stage models [31] - [48].

III. METHODOLOGY

The suggested methodology employs a hybrid framework with multiple stages to detect ransomware and zero-day malware, using machine learning and deep learning algorithms. At the first stage, correlation thresholding, Chi-square, mutual information, and variance ranking are used

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

to eliminate redundant and noisy attributes, thereby reducing feature dimensionality. Short- and long-term dependencies are then learned using a Bi-LSTM with a temporal CNN and a GRU-TCN hybrid. Such embeddings are further optimised using a residual TCN block with squeeze-and-excitation attention to highlight important behavioural patterns. Lastly, fused embeddings are categorised using a Hybrid XG-Fusion model with focal loss and an XGBoost meta-classifier to address class imbalance and enhance detection reliability.

A. Dataset Collection

The samples are 1,524 pieces of software, 582 ransomware, and 942 benign programs, which are dynamically tested by the Cuckoo Sandbox (2016). It includes samples from several ransomware families, including CryptoWall, TeslaCrypt, and PGPCODER, and each family has its own identifier. The dataset contains behavioural characteristics, API calls, registry entries, file operations, dropped file extensions, embedded strings, and SHA1 and MD5 hashes. All samples are categorised as ransomware or benign, enabling classification and evaluation in an experiment.

B. Data pre-processing

The raw data is purified and formatted into model training data. The names of the columns are homogenized and duplicates are eliminated to avoid bias. Rows with more than 30 per cent missing values are removed, and the remaining missing data points are replaced with 0, indicating binary behavioural characteristics. Min-Max scaling is used to normalise numerical attributes to the [0,1] range. The relevant feature groups, such as labels, family identifiers, and behavioural indicators, are extracted and encoded as categorical ransomware families using Label Encoder. This preprocessing guarantees clean, balanced, and well-scaled data, ensuring the model is trained successfully.

C. Dimensionality Reduction using Truncated SVD

Truncated SVD reduces the NSA/API feature space (high-dimensional) to 300-dimensional behavioural embeddings that capture essential ransomware trends while also mitigating noise and redundancy. This also enhances training speed, reduces memory usage, and improves generalisation. The hybrid feature matrix combines 300 embeddings and family encoding, followed by the label, resulting in a (1524, 302) dataset for classification. Let the original binary feature matrix be $X \in R^{nxm}$, where n is the list of samples and m is the list of binary API/string features. To isolate latent behavioural patterns and dimensionality reduction, Truncated Singular Value Decomposition (SVD) is applied:

$$X \approx U\Sigma V^{T} \qquad (1)$$

where $U \in R^{nxk}$ are the left singular vectors (sample embeddings), $\Sigma \in R^{kxk}$ contains the top k singular values, and $V \in R^{mxk}$ are the right singular vectors (feature directions). Here, k = 300.

The behaviour embedding matrix is:

$$\tilde{X} = U\Sigma \in R^{nxk} \qquad (2)$$

Each row $\tilde{x}_i \in R^k$ represents the low-dimensional behavioural flow vector of sample i.

Next, the hybrid feature matrix H incorporates categorical knowledge and labels:

$$H = \tilde{X} \parallel F \parallel Y \in R^{nx+k+c+1)} \qquad (3)$$

where $F \in R^{nxc}$ is the encoded malware family vector, and $Y \in R^{nx1}$ contains class labels.

1) Feature Entropy Computation

To quantify the variability and informativeness of each binary feature, the Shannon entropy is computed:

$$H(f_j) = -\Sigma\, p_{ij}\, log^2 p_{ij}, \qquad j = 1,\ldots,m \qquad (4)$$

where f_j is the j-th feature, b is the number of histogram bins (here 10), and p_{ij} is the normalized frequency of bin i for feature j. Features that have high-entropy form are more uncertain and contain more information about behaviour, and thus are essential for differentiating between ransomware and benign activity.

The resulting entropy vector is:

$$e = [H(f^1), H(f^2), \ldots, H(f^m)]^T \in R^{mx1} \qquad (5)$$

The top 15 high-entropy features are selected for further analysis, emphasizing the most informative aspects of malware behaviour in the feature space.

D. Feature Entropy Analysis and T-SNE Embedding

Shannon entropy quantifies the variability of API and string features and flags frequently abused operations (memory access, registry queries, and file) that leave discriminative behavioural fingerprints for ransomware classification.

After extracting the low-dimensional behavioural embedding $\tilde{X} \in R^{nxk}$ using Truncated SVD, we further explore the structure of ransomware and goodware using t-distributed Stochastic Neighbour Embedding (t-SNE). The t-SNE maps high-dimensional behaviour vectors into a 2-dimensional space for visualization:

$$Y = t - SNE(\tilde{X}) \in R^{nx2} \qquad (6)$$

where $Y_i = [y_i^{(1)}, y_i^{(2)}]$ represents the 2D embedding of the i-th sample. The t-SNE algorithm minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional pairwise similarities:

$$KL(P \,||\, Q) = \Sigma\, p_{ij}\, log\left(\frac{p_{ij}}{q_{ij}}\right) \qquad (7)$$

with p_{ij} and q_{ij} denoting probabilities of similarity in the original and embedded spaces, respectively. Hyperparameters include perplexity = 50 and learning rate = 200.

1) Behavioural Signal Differentiation

To identify the most discriminative features between ransomware (R) and goodware (G), we compute the absolute mean difference:

$$\Delta f_j = \left|\mu_j^{(A)} - \mu_j^{(G)}\right|, \qquad j = 1,\ldots,m \qquad (8)$$

samples, respectively. Features with the highest Δf_j highlight key behavioural signals that distinguish malware from

© Copyright: All rights reserved.

benign software. The top 20 features with the largest differences are selected for further analysis.

To ensure an unbiased evaluation and reliability of subsequent supervised learning, the dataset is stratified by class label and split:

$$X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}} = train_test_split(X, y, test_size = 0.2, stratify = y) \quad (9)$$

where $X \in R^{nxm}$ contains all API/STR numerical features, and $y \in \{0, 1\}^n$ represents the ransomware (1) / goodware (0) labels. The stratified split preserves class distribution in both training and testing sets.

The table I entropy score indicates the most informative features by their position in the list presented below. The API calls and strings are highly diverse across samples and can be more useful for modelling ransomware behaviour.

Table I: Top 15 High-Entropy Behaviour Features

Feature	Entropy
API: Nt Create Mutant	0.999980
API: Create Directory W	0.999939
API: Find First File Ex W	0.999899
API: Nt Read Virtual Memory	0.999756
API: Get System Directory W	0.999682
API: Virtual Protect Ex	0.999399
API: Reg Query Value Ex A	0.999223
API: Get System Info	0.999026
API: Co Initialize Ex	0.997702
API: Search Path W	0.997594
API: Virtual Free Ex	0.996374
STR:13946	0.995219
API: Size of Resource	0.992812
API: Nt Query Information File	0.990576
API: Set Unhandled Exception Filter	0.989688

Entropy values indicate randomness and behavioural richness. Higher scores represent more discriminative ransomware activity patterns.

Once behaviour embeddings are created, t-SNE with SVD is used to downsample them and enable visualisation. SVD is used to reduce the high-dimensional API/STR behaviour space to 300 dimensions, and t-SNE is used to project the resulting 300-dimensional representation into 2-D while preserving sample similarities. When perplexity is set to 50 and the learning rate to 200, ransomware and goodware clusters can be observed, behaviour patterns can be visualised, anomalies can be identified, and malicious and legitimate applications can be differentiated.

E. Feature Selection using Chi-Square Test and Mutual Information

The ransomware dataset had many redundant binary API and string features, which were narrowed down to the top 20 most discriminative features for each method using Chi-Square and Mutual Information. The data was divided into 80/20 train/test sets, keeping the classes balanced. Feature selection enhanced efficiency and accuracy, as t-SNE visuals revealed clear clusters of ransomware and goodware.

Table II: Top 20 Chi-Square Ranked Behaviour Features

Feature	Chi-Square Score
API: Nt Open File	26.4785
API: Get System Info	60.8478
API: Write Console W	18.7708
API: Nt Read Virtual Memory	175.4996
API: Remove Directory A	17.9521
API: Get Key State	76.7719
API: Nt Query Key	81.6566
API: Open Service W	10.7944
API: Enum Windows	28.5584
API: Get Volume Name for Volume Mount Point W	30.3615
API: Http Open Request A	9.0043
API: Get System Directory W	138.8257
API: Lookup Privilege Value W	93.1749
API: Get Disk Free Space Ex W	95.7169
API: Create Remote Thread	14.3597
API: Module32 First W	14.6917
API: Find Window Ex W	21.7496
API: Nt Enumerate Key	11.9983
API: Nt Free Virtual Memory	8.0398
API: Reg Query Info Key W	9.4941

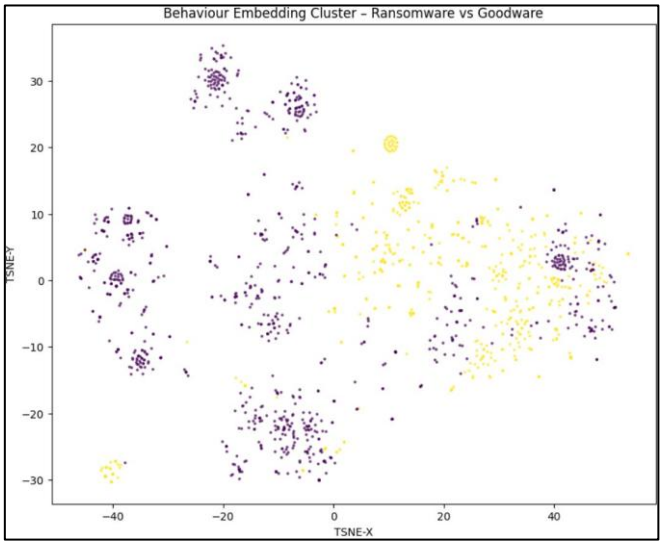
Higher Chi-square values indicate stronger relevance to ransomware behaviour patterns.

Table III: Top 30 Overall Common Contributing Features (Combined χ^2 + MI Ranking)

Rank Feature

1. API: Create Directory W
2. API: Co Initialize Ex
3. API: Set End of File
4. API: SearchPathW
5. API: Nt Read Virtual Memory
6. API: Set File Pointer
7. STR:6301
8. STR:8583
9. STR:3391
10. API: Nt Terminate Process
11. STR: 3550
12. API: Get System Directory W
13. API: Get File Attributes W
14. API: Size of Resource
15. API: Is Debugger Present
16. API: Nt Get Context Thread
17. API: Get File Size
18. API: Send Notify Message A
19. API: Nt Set Context Thread
20. API: Nt Duplicate Object
21. API: Lookup Privilege Value W
22. API: Nt Query Directory File
23. API: WSA Startup
24. API: Get Disk Free Space Ex W
25. API: Get Volume Path Names for Volume Name W
26. API: Get System Windows Directory W
27. API: Ole Initialize
28. STR:7283
29. API: Set Error Mode
30. API: Nt Set Information File

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.



[Fig.1: Behavioural Clustering of Ransomware]

Features were ranked using Chi-Square and Mutual Information (MI), and a combination of the two was used to determine the most relevant features for detecting ransomware. The 200 strongest features were then selected, with the 30 strongest listed in Table IV, including API: Create Directory W and API: Nt Read Virtual Memory, which capture critical behaviours such as file operations, memory access, and privilege management. This combined ranking reduces bias associated with using a single selection method and improves classification performance. The extracted features reflect ransomware behavioural traits, enabling detection of even zero-day malware without relying on known signatures.

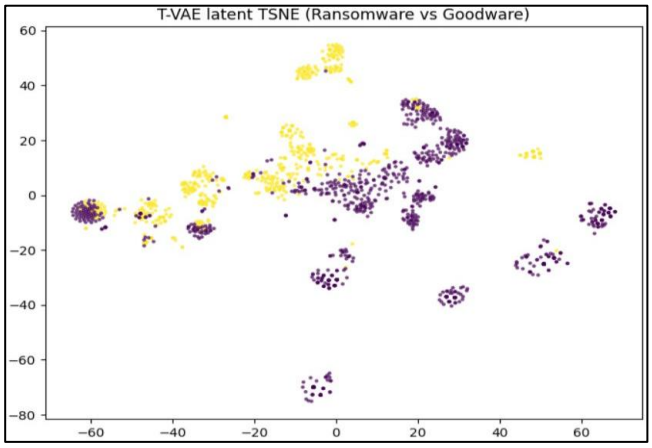
Characteristics such as file manipulation, memory access, and system resource usage are core behaviours of ransomware. They allow machine learning models to infer unseen threats by identifying uncharacteristic system activity. Detection is enhanced by concentrating on statistically/information relevant features. precision, minimises false positives, and reinforces proactive defence against dynamic ransomware.

F. Deep Learning-Based Embedding Feature Reduction

Stage-1: After Stage-1, 200 high-ranking features were identified by Chi-Square and MI, standardised, and divided into training and validation sets in a balanced manner to minimise redundancy and computation. Stage-2 A Transformer-based VAE was used to encode sequential behaviour to produce behavioural representations in a 128-dimensional latent space, which was optimised using MSE, MAE, SSIM, KL divergence, and Beta annealing to produce stable behavioural representations.

Table IV: Top 30 Overall Common Contributing Features (Combined χ^2 + MI Ranking)

Rank	Feature
1.	API: Create Directory W
2.	API: Co Initialize Ex
3.	API: Set End of File
4.	API: Search Path W
5.	API: Nt Read Virtual Memory
6.	API: Set File Pointer
7.	STR:6301
8.	STR:8583
9.	STR:3391
10.	API: Nt Terminate Process
11.	STR:3550
12.	API: Get System Directory W
13.	API: Get File Attributes W
14.	API: Size of Resource
15.	API: Is Debugger Present
16.	API: Nt Get Context Thread
17.	API: Get File Size
18.	API: Send Notify Message A
19.	API: Nt Set Context Thread
20.	API: Nt Duplicate Object
21.	API: Lookup Privilege Value W
22.	API: Nt Query Directory File
23.	API: WSA Startup
24.	API: Get Disk Free Space Ex W
25.	API: Get Volume Path Names for Volume Name W
26.	API: Get System Windows Directory W
27.	API: Ole Initialize
28.	STR:7283
29.	API: Set Error Mode
30.	API: Nt Set Information File



[Fig.2: T-VAE Latent Space after Transformer-VAE Feature Extraction]

Figure 2 shows t-SNE visualisation of T-VAE latent embeddings, revealing distinct ransomware (purple) and goodware (yellow) clusters. This approach captures general behavioural patterns, enabling classification of unseen and zero-day malware by compressing sequential API/STR features into a meaningful latent space.

1) Positional Encoding

Let $X \in R^{IxO}$ denote the batch of input samples with F features. To retain sequential ordering, each feature position pos ∈ [0, F − 1] is mapped to a d-dimensional vector using sinusoidal functions:

$$PE(pos,2i) = sin(pos / 10000^{2i/d}) \quad (10)$$

$$PE(pos,2i+1) = cos(pos / 10000^{2i/d}) \quad (11)$$

The positional encoding matrix $PE \in R^{Oxd}$ is added to the input embedding after projection:

© Copyright: All rights reserved.

$$X_{\text{proj}} = Linear(X) \in R^{\text{IxOxd}},$$
$$X_{\text{pos}} = X_{\text{proj}} + PE \qquad (12)$$

This allows the transformer to distinguish feature order, crucial for capturing behavioural sequences.

2) Transformer Encoding and Latent Variable Modelling

The transformer encoder maps X_{pos} to hidden representations $H \in R^{Ixd}$:

$$H = TransformerEncoder(X_{\text{pos}}) \qquad (13)$$

Mean pooling across sequence length reduces H to a single vector per sample. Then, the encoder predicts the parameters of the latent distribution:

$$\mu, log\sigma^2 = Linear(H) \qquad (14)$$

The latent vector Z is sampled via the reparameterization trick:

$$Z = \mu + \varepsilon \cdot exp(½\, log\sigma^2), \qquad \varepsilon \sim N(0, I) \qquad (15)$$

3) Decoder and Reconstruction

The decoder reconstructs the original feature vector from Z:

$$\hat{X} = g\varphi(Z) = MLP(Z) \qquad (16)$$

The training minimizes a composite loss:

$$L = MSE(X, \hat{X}) + MAE(X, \hat{X}) + \beta\, D\text{□□}(q\theta(Z|X)|p(Z)|) \qquad (17)$$

Transformer-VAE compresses behavioural characteristics and preserves latent patterns that reflect the presence of undetected, zero-day ransomware. With knowledge of the distributions of normal and malicious API/STR sequences, positional encoding, and transformer layers, it identifies fine-grained long-range dependencies. The model, together with Stage-1 feature selection, is an informative, signal-oriented, and thus robust approach to real-time zero-day ransomware detection.

Table V: Transformer-VAE Hyperparameters for Behavioural Embedding

Parameter	Value	Description	Notes
Latent Dimension	128	Size of latent embedding	Captures compressed features
D Model	256	Transformer model dimension	Controls representation capacity
Heads	8	Number of attention heads	Multi-head attention
Layers	3	Transformer encoder layers	Depth of model
Batch Size	32	Training batch size	Stable gradients
Learning Rate	5×10^{-4}	Optimizer step size	AdamW optimizer
Epochs	120	Training iterations	Ensures convergence
Beta Start/End	0.001/1.0	KL warm-up	Gradual regularization

Source: Tuned hyperparameters ensure stable training and robust latent embeddings for ransomware detection.

Table V lists the key hyperparameters of the Transformer-VAE model. These parameters control the latency dimension, the depth of the transformer, the number of attention heads, learning dynamics, and Beta annealing. Their representation of behaviour proves effective through proper tuning, ensuring stable training, correct reconstruction, and high-quality latent embeddings, and they are effective in defence against known and zero-day ransomware exploits.

Stage-2 Transformer-VAE represents samples in a 32-64-dimensional latent space, preserving feature dependencies and temporal patterns. These mean vectors (μ) are used as embeddings to achieve reproducibility, analysis, and 2D visualization and high-quality zero-day ransomware detection based on the behavioral clustering.

Following the Transformer-VAE encoding, we obtain a compact latent representation $Z \in R^{nxH}$ for all n samples, where L is the latent dimension (32-64). Let the VAE encoder produce mean μ_i and log-variance $log\sigma^2$ for sample i:

$$\mu_i, log\sigma^2 = f\theta(X_i), \qquad i = 1, \ldots, n \qquad (18)$$

The latent vector is sampled via the reparameterization trick:

$$Z = \mu + \varepsilon \cdot exp(½\, log\sigma^2), \qquad \varepsilon \sim N(0, I) \qquad (19)$$

For feature extraction and downstream tasks, the mean vector μ_i is used as the deterministic latent embedding:

$$Z^{\text{embedd}_i^{kG}} = \mu_i \qquad (20)$$

The latent embeddings preserve feature dependencies and temporal behaviour, as captured by the transformer layers and positional encoding from Stage-2.

To visualize and analyze these embeddings, a dimensionality reduction using t-SNE is applied:

$$Z^{2D} = t - SNE(Z^{\text{embedd}_i^{kG}}) \qquad (21)$$

This maps the high-dimensional latent space to 2D, highlighting clusters of ransomwares and goodware, and ensuring generalization to zero-day ransomware through behaviourally-informed embeddings.

G. Multistage Classification using ML Models

Transformer-VAE latent embeddings capture temporal and feature interactions that differentiate ransomware, goodware, and variants of zero-day attacks. These embeddings were partitioned into 80:20 and used to benchmark baseline classifiers (RF, XGBoost, SVM, LR, KNN). Findings demonstrate that deep latent representations are more effective than classical models, which justifies using the Transformer-VAE as a solid foundation for detecting real-world ransomware.

H. Proposed Deep Hybrid Detector for Zero-Day Ransomware

The four-stage deep intelligence pipeline identifies the presence of ransomware and zero-day attacks by filtering features, temporal knowledge, and multi-scale behaviours. Models are inhibited as Bi-LSTM + GRU + TCN + Residual stack + SE attention + Boost fusion to address long-term flows, short anomalies, and hierarchical patterns. They have the combined effect of facilitating high-confidence

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

detection with enhanced sensitivity, resistance to obfuscation, and concept drift.

1) Bi-LSTM + Temporal CNN Hybrid Model

Transformer-VAE latent embeddings, combined with Bi-LSTM and multi-kernel CNN, are used in the Deep Hybrid Detector to identify fine-tuned ransomware behaviours, such as zero-day variants, and it achieves good detection performance, outperforming classical baselines across all significant metrics.

Input Congenerative Transformer-VAE positional encodings constitute the input data, X. The Bi-LSTM, on the other hand, captures long-range sequential relationships. In contrast, the multi-kernel CNN captures short-term bursts. They combine their outputs and feed them into a complete network trained with a binary cross-entropy loss to identify zero-day ransomware based on invisible timing and behavioural patterns.

$$Z = [z^1, z^2, \ldots, z_n] \in R^{nxи} \qquad (22)$$

where n is the number of samples and L is the latent dimension. These embeddings encode both temporal patterns and feature dependencies.

Table VI: Hyperparameters of the Deep Hybrid Detector (Bi-LSTM + Temporal CNN)

Hyperparameter	Value	Description	Notes
LSTM Hidden Size	64	Number of units in Bi-LSTM	Bidirectional
LSTM Layers	2	Number of stacked LSTM layers	Dropout applied
CNN Channels	32	Number of filters per kernel	Multi-kernel CNN
Kernel Sizes	3,5,7	Temporal convolution kernel sizes	Captures burst patterns
Dropout Rate	0.4	Dropout probability	Prevent overfitting
FC Layers	4	Number of fully connected layers	Deep classification
Learning Rate	0.001	Optimizer learning rate	Adam optimizer
Batch Size	2	Mini-batch size	Small batch for stability
Epochs	100	Training iterations	Full convergence

Source: Configurations used for training the Deep Hybrid Detector.

To retain the sequential order of features, positional encoding is added:

$PE(pos,2i) = sin(pos / 10000^{2}i/d_{mo}{}^{d}{}_{e}\ell)$ (23)

$PE(pos,2i+1) = cos(pos / 10000^{2}i/d_{mo}{}^{d}{}_{e}\ell)$ (24)

The final input to the model becomes:

$X_{inp}{}^{u}{}_{t} = Z + PE$ (25)

ensuring the model captures temporal dependencies in the latent sequence. The Bi-LSTM processes the sequential embeddings bidirectionally:

$$\overrightarrow{h_t} = LSTM^{fWd}\left(X_t, \overrightarrow{h_t^{-1}}\right) \qquad (26)$$

$$h_t = LSTM^{bWd}(X_t, \overleftarrow{h}_{t+1}) \qquad (27)$$

$$h^{LLLL} = [\overrightarrow{h}_t ; \overleftarrow{h}_t] \qquad (28)$$

Layer normalization is applied:

$$\hbar^{LLLL} = LayerNorm(h^{LLLL}) \qquad (29)$$

Temporal patterns and burst behaviours are extracted via multiple 1D convolutions:

$$f^k = ReLU\left(Conv1D^k(X_{inp}{}^{u}{}_{t})\right), \quad k \in \{3,5,7\} \qquad (30)$$

The outputs are pooled and concatenated:

$f^{cOO} = Linear\ (Concat(AvgPool(f_3), AvgPool(f_5), AvgPool(f_7)))$ (31)

The LSTM and CNN features are fused:

$$f^{fu}{}_{sion} = [\hbar^{LLLL} ; f^{cOO}] \qquad (32)$$

This fused vector is processed through a deep, fully connected network:

$$\hat{y} = \sigma(FC(f^{fu}{}_{sion})) \qquad (33)$$

Supervised training uses binary cross-entropy:

$$L = -\left(\frac{1}{n}\right) \Sigma\, y_i\, log\, \hat{y}_i + (1 - y_i) log(1 - \hat{y}_i) \qquad (34)$$

GRU-TCN Hybrid learns latent VAE Transformer dependencies and burst patterns. Bi-GRU trains on forward- and reverse-ransomware patterns, and TCN trains on high-frequency local patterns. The combination of features is normalised by a fully connected network with ReLU, dropout, and Gaussian noise, enabling powerful detection of unseen zero-day ransomware with small, information-rich representations.

2) GRU-Temporal CNN Hybrid Model

The GRU-TCN Hybrid model is designed to capture both long-term temporal dependencies and short-term burst patterns in latent embeddings obtained from the Transformer-VAE stage. It consists of two parallel branches: a bidirectional GRU and a Temporal Convolutional Network (TCN), followed by a fusion layer and a deep classifier.

GRU Branch

The input latent vector $X \in R^{IxO}$ is treated as a sequence of length 1 with F features. The bidirectional GRU computes hidden states h_t for each time step:

$$h^{GRU} = BiGRU(X), \quad h^{GRU} \in R^{Ix2H} \qquad (35)$$

where H is the hidden dimension, the last hidden state is used as the temporal feature representation:

$$h^{GRUfe}{}_{at} = h^{GRU}[:, -1, :] \qquad (36)$$

TCN Branch

Simultaneously, the TCN applies a 1D convolution over the latent features:

$$h^{TcO} = ReLU(Conv1D(X)), \quad h^{TcO} \in R^{Ixc} \qquad (37)$$

where C denotes the number of TCN channels, global average pooling reduces it to a compact vector:

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

$$h_t{}^{c}{}_{n}{}^{fe}{}_{at} = (1/F)\,\Sigma\, h^{TcO},_{i} \qquad (38)$$

Fusion and Classification

The GRU and TCN features are concatenated and normalized:

$$Z = BatchNorm([h^{GRUfe}{}_{at} \,||\, h_t{}^{c}{}_{n}{}^{fe}{}_{at}]) \qquad (39)$$

Finally, Z is passed through fully connected layers with ReLU and dropout to produce the probability of ransomware detection:

$$\hat{Y} = \sigma(FC(Z)), \qquad \hat{Y} \in [0,1] \qquad (40)$$

This architecture enables effective detection of zero-day ransomware, as GRU captures temporal dependencies, TCN identifies short-term bursts, and their fusion integrates both behaviours for robust classification.

3) Hybrid Enhanced + SE-Residual-TCN Architecture

The Hybrid-Enhanced model uses Bi-GRU to obtain sequential latent features, deep Residual-TCN to capture multi-scale temporal patterns, and SE-Block attention to obtain adaptive channel weighting. GRU catches changes in behaviour that emerge, whereas TCN catches short bursts, such as rapid encryption or privilege escalation. SE attention fuses and refines outputs to highlight malicious content while suppressing benign noise. Finally, a multilayer classifier forecasts ransomware, enabling highly effective zero-day detection. Let latent feature input be $X \in R^{IxO}$. Noise augmentation improves robustness:

$$X' = X + \varepsilon, \qquad \varepsilon \sim N(0, 0.012^2) \qquad (41)$$

GRU Branch: Long-Term Behaviour Modelling

Input is reshaped into sequential form:

$$h^{GRU} = BiGRU(X') \in R^{Ix2H}, \quad h^{GRUfe}{}_{at} = h^{GRU}[:,-1,:] \qquad (42)$$

Residual TCN Branch: Burst Pattern Extraction

TCN consists of stacked residual blocks:

$$y^1 = f\left(BN\left(Conv1D(X')\right)\right), \quad y^2 = f\left(BN\left(Conv1D(y^1)\right)\right) \qquad (43)$$

Residual formulation:

$$h_t{}^{c}{}_{n} = f(y_2 + X'), \quad h_t{}^{c}{}_{n}{}^{fe}{}_{at} = (1/F)\,\Sigma\, h_t{}^{c}{}_{n,i} \qquad (44)$$

SE Attention: Feature Re-Weighting

Squeeze and excitation adjust feature contribution:

$$s = \sigma\left(W^2 ReLU\left(W^1\left[h^{GRUfe}{}_{at} \,|h_t{}^{c}{}_{n}{}^{fe}{}_{at}|\right]\right)\right) \qquad (45)$$

$$Z = s \odot ([h^{GRUfe}{}_{at} \,||\, h_t{}^{c}{}_{n}{}^{fe}{}_{at}]) \qquad (46)$$

Final Prediction

$$\hat{Y} = \sigma(MLP(Z)) \qquad (47)$$

4) Hybrid XG-Fusion Model (GRU + Dilated-TCN + Attention + Focal Loss)

The proposed Hybrid XG-Fusion model integrates Bidirectional-GRU, Dilated Residual Temporal Convolution Networks, and Attention-based fusion for high-precision anomaly detection. Let the latent input be $x \in R^{Ixd}$. To capture temporal dependencies, we reshape x as a sequence $x_t \in R^{Ix1xd}$, processing it through a bidirectional GRU:

The latent input is reshaped to sequential form $X_t \in R^{Ix1xd}$ and processed through a Bi-GRU:

$$\overrightarrow{h_t} = GRU^{f}(X_t), \qquad \overleftarrow{h_t} = GRU^{b}(X_t) \qquad (48)$$

$$H_t = [\overrightarrow{h_t} \,||\, \overleftarrow{h_t}] \in R^{Ix2H} \qquad (48)$$

The TCN stack with dilation factors d = {1, 2, 4} learns hierarchical temporal dependencies:

$$Z_2 = \sigma(BN(Conv(Z_1;\, k=3, d=2))), \quad Z_3 = \sigma(BN(Conv(Z_2;\, k=3, d=4))) \qquad (51)$$

The residual formulation is expressed as:

$$TCN_r{}^{e}{}_{s}(X_t) = Z^3 + S(X_t) \qquad (52)$$

Global average pooling compresses the feature maps:

$$T = \left(\frac{1}{n}\right)\Sigma\, Z^{(i)} \in R^{Ixc} \qquad (53)$$

Attention learns relevance scores per timestep:

$$e_t = W_a H_t, \qquad \alpha_t = \frac{exp(e_t)}{\Sigma^k\, exp(e^k)} \qquad (54)$$

Final attended GRU representation:

$$H_{att} = \Sigma_t\, \alpha_t\, H_t \qquad (55)$$

$$F = [H_{att} \,||\, T] \in R^{Ix(2H+c)} \qquad (56)$$

$$\hat{y} = \sigma(W^2 ReLU(W^1 F + b^1) + b^2) \qquad (57)$$

To handle class imbalance, we optimize Focal Loss instead of BCE:

$$L^{f}{}_{o}{}^{c}{}_{a}\ell = \alpha(1 - p_t)^{x}(-log(p_t)) \qquad (58)$$

Were

$$p_t = \hat{y}t + (1 - \hat{y})(1 - t), \quad \alpha = 0.75, \gamma = 3.0 \qquad (59)$$

The end-to-end optimization is: (60)

Because H_{att} encodes long-range, attention-weighted temporal context and T captures multi-scale burst patterns, the fused representation F emphasises behavioural signatures rather than family-specific indicators, enabling robust generalisation to unseen (zero-day) ransomware variants.

IV. RESULTS AND DISCUSSION

A comparison of five machine learning (ML) and four deep learning (DL) models reveals the usefulness of multi-stage analytics for detecting ransomware, even in the case of zero-day attacks. XGBoost, random forest, and KNN had around

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

93.44 per cent accuracy, while logistic regression and SVM failed on dynamic execution signatures. The GRU-TCN architecture is one of the most successful DL models, achieving high recall and accuracy (94.10%). Hybrid-XG-Fusion yielded the largest ROC-AUC (0.9649) to distinguish between benign and malicious samples. The Bi-LSTM + TCN architecture reduced false alarms, and the Hybrid SE-Residual-TCN architecture was the most effective at identifying low-frequency anomalies.

A. Performance of Machine Learning Classifiers

XGBoost and Random Forest, with 93.44 per cent and 0.944 precision, captured the nonlinear pattern of execution well, though with moderate recall (0.8793), thereby under-reporting rare ransomware patterns. KNN was also as accurate as this and gave the highest F1-score (0.9115), but with better recall (0.8879), albeit at the cost of increased computational complexity. The use of Logistic Regression was inexpensive and quick, with an accuracy of 91.1 per cent and an ROC-AUC of 0.9676, but failed to address or handle complex and polymorphic patterns. SVM had a poorer recall (0.8017) and was poor at generalising nonlinear behaviour. Overall, the performance of tree-based and KNN models is good as baselines; temporal deep learning is required to detect zero-days adaptively.

Table VII summarises the performance of five ML models on 1219 training and 305 testing samples. XGBoost and Random Forest achieved the highest accuracy (0.9344), whereas KNN delivered the best F1-score (0.9115). SVM shows comparatively lower recall, indicating that it misses malicious samples under high-variance behaviour.

Table VII: Performance Comparison of ML Classifiers on Ransomware Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
XG Boost	0.9344	0.9444	0.8793	0.9107	0.9828
Random Forest	0.9344	0.9444	0.8793	0.9107	0.9826
Logistic Regression	0.9114	0.9082	0.8534	0.8800	0.9676
KNN	0.9344	0.9363	0.8879	0.9115	0.9650
SVM	0.9049	0.9393	0.8017	0.8651	0.9536

B. Deep Learning Model Performance

Model-1 (Bi-LSTM+TCN) achieved an accuracy of 0.918 and a precision of 0.969 but exhibited lower recall (0.8103). Model-2 (GRU+TCN) delivered the best overall performance, achieving an accuracy of 0.9410, an F1 score of 0.9196, and a recall of 0.8879. Model-3 (Residual-TCN+SE) improved ROC-AUC (0.9607) but showed reduced recall (0.8276). Model-4 (GRU+ Dilated- TC N+ Attention+ Focal Loss) balanced recall (0.8620) and ROC-AUC (0.9649), enhancing minority class detection. The integration of CNN, residual connections, and attention mechanisms strengthened feature learning. Overall, Model-2 achieved the strongest performance, while Model-4 provided the most effective real-world trade-off.

Table VIII: Comparative Performance of Proposed Deep Learning Models

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Bi-LSTM+TCN	0.9180	0.9691	0.8103	0.8826	0.9553
GRU+TCN	0.9410	0.9537	0.8879	0.9196	0.9587
Hybrid-Enhanced SE-TCN	0.9115	0.9320	0.8276	0.8767	0.9607
Hybrid XG-Fusion	0.9311	0.9523	0.8620	0.9049	0.9649

Table VIII shows that Model 2 (GRU+TCN) achieved the highest accuracy (0.9410) and recall, while Model 4 (Hybrid XG-Fusion) attained the best ROC-AUC (0.9649), excelling in zero-day detection. Generally, hybrid deep learning models are more effective than classical machine learning methods because they use sequential memory, multi-scale convolutions, and attention mechanisms to detect ransomware efficiently.

V. CONCLUSION

This paper presents a robust multi-stage hybrid pipeline for the detection of ransomware and zero-day malware, designed to overcome the inherent limitations of traditional signature-based approaches. By focusing on behavioural analysis, end-to-end learning, and generalisation, the proposed framework effectively identifies previously unseen and evolving threats across heterogeneous computing environments.

The architecture integrates statistical feature selection, deep latent embedding generation, attention-based refinement, and hybrid ensemble classification. The initial feature optimisation stage significantly reduces noise in high-dimensional behavioural data while preserving discriminative patterns, thereby enabling efficient downstream learning. Deep sequential models, including Bi-LSTM with Temporal CNN and GRU-TCN hybrids, capture complex temporal dependencies and abnormal execution patterns without relying on known signatures. Furthermore, the Hybrid-XG-Fusion model enhances detection robustness by integrating attention mechanisms and focal loss optimization to address class imbalance and rare attack scenarios. The use of XGBoost as a meta-classifier strengthens predictive reliability through gradient-boosted ensemble learning. Experimental evaluations demonstrate superior performance in terms of accuracy, precision, recall, F1-score, and ROC-AUC. Notably, the GRU-TCN model achieved strong overall performance, while the Hybrid-XG-Fusion model attained the highest ROC-AUC, highlighting its effectiveness in zero-day detection. The modular and hierarchical design improves discrimination between malicious and benign behaviours, reduces false positives, and enhances the identification of rare or emerging ransomware variants. Additionally, the framework ensures scalability and adaptability in diverse and distributed computing environments. Future work may focus on real-time deployment, integration with cloud and edge infrastructures, and extension to a broader range of malware families. Incorporating reinforcement learning for adaptive threat

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

mitigation and further optimising attention mechanisms could enhance proactive defence capabilities. Overall, this study demonstrates that behavioural embeddings, combined with hybrid fusion models, provide a promising foundation for next-generation intelligent malware detection systems.

DECLARATION STATEMENT

As the article's author, I must verify the accuracy of the following information after aggregating input from all authors.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted objectively and without external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. H. Khalid and W. A. Alawsi, "Hybrid CNN-LSTM Model for Real-Time Detection of Zero-Day Attacks in Heterogeneous IoT Networks," Journal of Information Hiding and Multimedia Signal Processing, vol. 16, no. 3, pp. 1058-1069, 2025. Available: https://bit.kuas.edu.tw/2025/vol16/N3/18.JIHMSP-250701.pdf
2. S. Adel, A. Sarah, F. M. Talaat, and M. M. Saafan, "Emerging AI threats in cybercrime: a review of zero-day attacks via machine, deep, and federated learning," Knowledge and Information Systems, vol. 67, no. 11, pp. 10951-10987, 2025, DOI: http://doi.org/10.1007/s10115-025-02556-6.
3. H. N. Alshareef, "Current Development, Challenges, and Future Trends in Cloud Computing: A Survey," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, pp. 329-338, 2023, DOI: http://doi.org/10.14569/IJACSA.2023.0140337.
4. Y. Guo, "A review of machine learning-based zero-day attack detection: Challenges and future directions," *Computer Communications*, vol. 198, pp. 175–185, 2023, DOI: http://doi.org/10.1016/j.comcom.2022.11.001.
5. A. De Paola, S. Drago, P. Ferraro, and G. Lo Re, "Detecting Zero-Day Attacks under Concept Drift: An Online Unsupervised Threat Detection System," Journal of Network and Systems Management, vol. 32, no. 3, p. 54, 2024, DOI: http://doi.org/10.1007/s10922-024-09821-z.
6. Mupila, F. K., Gupta, H., & Bhardwaj, A. (2023). AI-driven adaptive access control in multi-cloud environments: A cognitive security framework. *Journal of Information Security and Applications, 10*(28), 1–15. DOI: https://doi.org/10.52783/jisem.v10i28s.4345.
7. A. Yousif, D. Siale, Q. Mustafa, Z. Hassan, M. Fakhrulddin, and A. Sedeeq, "Enhancing Large-Scale Network Security with a VGG-Net-Based DCNN," Journal of Resilience and Cyber Security, vol. 6, no. 3, pp. 1316-1331, 2025, DOI: http://doi.org/10.18196/jrc.v6i3.25169.
8. I. Basim, A. Fakhfakh, and A. M. Makhlouf, "A Hybrid Deep Learning Approach for Adaptive Cloud Threat Detection with Integrated CNNs and RNNs," Journal of Resilience and Cyber Security, vol. 6, no. 3, pp. 1129-1141, 2025, DOI: http://doi.org/10.18196/jrc.v6i3.25618.
9. S. K. Sundaramurthy, N. Ravichandran, A. C. Inaganti, and R. Muppalaneni, "AI-Driven Threat Detection: Leveraging Machine Learning for Real-Time Cybersecurity in Cloud Environments," Artificial Intelligence and Machine Learning Review, vol. 6, no. 1, pp. 23-43, 2025, DOI: http://doi.org/10.69987/AIMLR.2025.60104.
10. I. H. Rakib, "Deep Learning for Cybersecurity: Recent Advances in Threat Detection and Attack Mitigation," International Journal of Cyber-Security and Digital Forensics, vol. 12, no. 2, pp. 72-81, 2025. [Online]. Available: https://ijcsdf.org
11. R. Morshedi, "A Comprehensive Review of Deep Learning Techniques for Anomaly Detection in IoT Networks," Engineering Reports, vol. 7, no. 4, p. e70415, 2025, DOI: http://doi.org/10.1002/eng2.70415.
12. H. Kotb, E. Badr, and F. Sakr, "Recent Studies and Review on Detection of Cyber Threats in Cloud Security Using Artificial Intelligence," Journal of Cloud Computing, vol. 14, no. 1, pp. 13-31, 2025, DOI: http://doi.org/10.21608/jocc.%202025.446635.
13. M. R. Naeem, R. Amin, and M. Farhan, "Cyber security Enhancements with Reinforcement Learning: A Zero-Day Vulnerability Identification Perspective," PLoS ONE, vol. 20, no. 5, p. e0324595, 2025. DOI: http://doi.org/10.1371/journal.pone.0324595.
14. A. Alqhatani and S. Mehmood, "Deep Memory for Deep Threats: A Novel Architecture Combining GRUs and Deep Learning," PLOS ONE, vol. 20, no. 6, p. e0332752, 2025, DOI: http://doi.org/10.1371/journal.pone.0332752.
15. A. J. Aparcana-Tasayco, X. Deng, and J. H. Park, "A Systematic Review of Anomaly Detection in IoT Security," Symmetry, vol. 17, no. 2, p. 201, 2025, DOI: http://doi.org/10.3390/sym17020201.
16. N. Jayakrishna and N. N. Prasanth, "A Hybrid Deep Learning Model for Detection and Mitigation of DDoS Attacks in VANETs," Scientific Reports, vol. 15, no. 1, art. 34170, 2025, DOI: https://doi.org/10.1038/s41598-025-15215-1
17. J. Wu, S. Fu, and M. Sarabi, "Introducing a Hybrid Intrusion Detection Method for IoT-Cloud Environments Based on ResNeXt and Improved Ebola Optimization Search Algorithm," Scientific Reports, vol. 15, art. 37612, 2025, DOI: http://doi.org/10.1038/s41598-025-21408-5.
18. A. V. Nagarjun and S. Rajkumar, "Quantum Deep Learning-Enhanced Ethereum Blockchain for Cloud Security," Blockchain and Cryptocurrencies, vol. 7, no. 4, pp. 321-341, 2025. [Online]. Available: https://blockchainjournal.org
19. R. Amin, A. Costanzo, L. R. Alzabin, A. Aqdus, S. F. Kamarulzaman, and A. H. Alshehri, "An Efficient Federated Learning-Based Defence Mechanism for Software-Defined Network Cyber Threats Through Machine Learning Models," Scientific Reports, vol. 15, art. 41390, 2025, DOI: http://doi.org/10.1038/s41598-025-25345-1.
20. C. Christy, A. Nirmala, A. M. O. Teena, and A. I. Amali, "Machine Learning Based Multi-Stage Intrusion Detection System and Feature Selection Ensemble Security in Cloud Assisted Vehicular Ad Hoc Networks," Scientific Reports, vol. 15, art. 27058, 2025, DOI: http://doi.org/10.1038/s41598-025-96303-0.
21. M. Zahid and T. Singh, "Enhancing Cybersecurity in IoT Systems: A Hybrid Deep Learning Approach," Springer International Publishing, 2024, DOI: http://doi.org/10.1007/s43926-025-00156-y.
22. N. A. Memon, M. Sultana, E. Afshan, A. Siddiqui, M. Murtaza, and N. A. Memon, "Investigating the Effectiveness of Artificial Intelligence in Detecting Zero-Day Attacks," Lecture Notes in Networks and Systems, vol. 758, pp. 804-817, 2024, DOI: http://doi.org/10.1007/978-981-99-8739-4_67.
23. S. Ahmed, T. N. Usmani, D. I. Ahmed, R. A. Zafar, M. Z. Hussain, and M. Z. Hasan, "Robust and Explainable Hybrid Deep Learning Model for Real-Time Zero-Day Botnet Detection," Lecture Notes in Networks and Systems, vol. 758, pp. 948-961, 2024. DOI: http://doi.org/10.1007/978-981-99-8739-4_78.
24. K. Nisioti and G. Spathoulas, "Systematic Review of Approaches and Solutions for Combating Zero-Day Vulnerabilities," ACM Computing Surveys, vol. 56, no. 12, pp. 102071-102091, 2024, DOI: http://doi.org/10.1145/3678789.
25. S. Kumar, M. A. Bari, and R. Sharma, "Zero-Day Attack Detection in Multi-Tenant Cloud Environments," International Journal of Information Security and Cyber Forensics, vol. 38, no. 3, pp. 427-446, 2024. [Online]. Available: https://ijiscf.org
26. A. D. Vibhute and V. Nakum, "Deep Learning-Based Network Anomaly Detection and Classification in Cloud Environments," Procedia Computer Science, vol. 232, pp. 1636-1645, 2024, DOI: http://doi.org/10.1016/j.procs.2024.01.161.
27. S. Najafli, A. Toroghi Haghighat, and B. Karasfi, "A Novel Reinforcement Learning-Based Hybrid Intrusion Detection System on Fog-to-Cloud Computing," The Journal of Supercomputing, vol. 80, pp. 26088-26110, 2024, DOI: http://doi.org/10.1007/s11227-024-06417-x.
28. M. Aliakbarisani, P. Ghasemi, and A. Bakhshi, "Few-Shot Network Attack Detection with Metric-Fused Prototypical Networks," Computers & Security, vol. 145, p. 103456, 2024, DOI: http://doi.org/10.1016/j.cose.2024.103456.
29. K. Kowsalyadevi, "Federated Learning-based Routing Vulnerability Analysis and Attack Detection," International

© Copyright: All rights reserved.

Journal of Internet of Things and Web Services, vol. 17, no. 2, pp. 123-139, 2024.
DOI: http://doi.org/10.22266/ijies2024.0430.34.
30. M. Atakari, S. Doroudi, and C. Ardakani, "Deep Learning-Based Security Model for ERP-Integrated IoT in Manufacturing," Journal of Manufacturing Technology Management, vol. 35, no. 3, pp. 90-98, 2024, DOI: http://doi.org/10.22266/ijies2024.0430.34.
31. M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, and A. H. Khan, "Enhancing Intrusion Detection: A Hybrid Machine and Deep Learning Approach," Journal of Cloud Computing, vol. 13, no. 1, p. 25, 2024, DOI: http://doi.org/10.1186/s13677-024-00685-x.
32. A. Almadhor, A. Altalbe, I. Bouazzi, and A. Al Hejaili, "Strengthening Network DDOS Attack Detection in IoT Environments with Federated XAI," IEEE Access, vol. 12, pp. 35421-35436, 2024, DOI: http://doi.org/1.10.1109/ACCESS.2024.3421234.
33. N. Algarica, I. Winterburn, J. Penrose, and K. Greythorne, "Cryptographic Behavioural Signatures for Ransomware Detection: A Novel Approach," Journal of Cybersecurity and Privacy, vol. 4, no. 3, pp. 401-421, 2024. [Online]. Available: https://www.mdpi.com/2624-800X/4/3
34. N. Sahani, R. Zhu, J.-H. Cho, and C.-C. Liu, "Machine Learning-Based Intrusion Detection for Smart Grid Computing: A Survey," ACM Transactions on Cyber-Physical Systems, vol. 7, no. 2, pp. 1-31, 2023. DOI: http://doi.org/1.10.1145/3578366.
35. Z. He and H. Sayadi, "Image-Based Zero-Day Malware Detection in IoMT Devices: A Hybrid AI Method," Sensors, vol. 23, no. 18, p. 7731, 2023, DOI: http://doi.org/10.3390/s23187731.
36. D. V. Talati, "Enhancing Multi-Cloud Security with Quantum-Resilient AI for Anomaly Detection," Applied Sciences, vol. 13, no. 2, pp. 629-638, 2023, DOI: http://doi.org/10.3390/app13020629.
37. S. Ali, G. Adeem, S. U. Rehman, S. Hussain, and S. S. Raza, "Computational Intelligence Approaches for Zero-Day Attack Detection," Journal of Ambient Intelligence and Smart Environments, vol. 14, no. 6, pp. 27-36, 2022, DOI: http://doi.org/10.3233/AIS-200622.
38. I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "IoT Security Intelligence: Comprehensive Overview and Machine Learning Solutions," Preprints, 2022, DOI: http://doi.org/10.20944/preprints202203.0087.v1.
39. B. I. Farhan, B. I. Farhan, and A. D. Jasim, "Survey of Intrusion Detection Using Deep Learning in IoT," International Journal of Cyber Security Management, vol. 3, no. 1, pp. 1-18, 2022. DOI: http://doi.org/10.52866/ijcsm.%202022.01.01.009.
40. S. S. Mahadik, "Edge-HetIoT Defence Against DDoS Attack Using Learning Techniques," IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 3876-3890, 2022. [Online]. Available: https://ieeexplore.ieee.org
41. H. A. Alamri and V. Thayananthan, "Analysis of Machine Learning for Securing Software-Defined Networking," Procedia Computer Science, vol. 194, pp. 229-236, 2021. DOI: http://doi.org/10.1016/j.procs.2021.10.078.
42. J. Uddoh, D. Ajiga, B. P. Okare, and T. D. Aduloju, "AI-Based Threat Detection Systems for Cloud Infrastructure," IEEE Transactions on Services Computing, vol. 14, no. 1, pp. 61-67, 2021, DOI: http://doi.org/10.1109/TSC.2020.3041145.
43. K. I. Iyer, "From Signatures to Behaviour: Evolving Strategies for Next-Generation Intrusion Detection," IEEE Communications Magazine, vol. 59, no. 6, pp. 165-171, 2021, DOI: http://doi.org/10.1109/MCOM.2021.9459216.
44. M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proencaa, "Adversarial Deep Learning Detection and Defence Against DDoS in SDN," IEEE Access, vol. 9, pp. 156-167, 2021, DOI: http://doi.org/10.1109/ACCESS.2021.3078956.
45. J. Iong and Z. Chen, "IoT Authentication and Access Control by Hybrid Deep Learning Method," International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, pp. 236-245, 2021. [Online]. Available: https://ijacsa.thesai.org
46. H. Hindy, R. Atkinson, C. Tachtatzis, J. Colin, E. Bayne, and X. Bellekens, "Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection," Electronics, vol. 9, no. 10, p. 1684, 2020, DOI: http://doi.org/10.3390/electronics9101684.
47. I. Alrashdi and A. Alqazzaz, "FBAD: Fog-Based Attack Detection for IoT Healthcare in Smart Cities," in Proceedings of the 12th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 515-522, 2020. DOI: https://doi.org/10.1109/UEMCON47517.2019.8992963
48. B. A. S. Al-rimy, M. A. Maarof, Y. A. Prasetyo, Z. M. Shaid, A. Fadillah, and M. Ariffin, "Zero-Day Aware Decision Fusion-Based Model for Crypto-Ransomware Detection," International Journal of Advanced Computer Science and Applications, vol. 9, no. 1, pp. 82-88, 2018, DOI: http://doi.org/1.10.14569/IJACSA.2018.090111.

AUTHOR'S PROFILE



Madhan Mohan Reddy Chinthala, received the M.S. degree in Computer Science from the University of Dayton, USA, and the B.Tech. degree in Information Technology from J.B. Institute of Engineering and Technology, India. He is currently working as a Network Engineer, with professional experience in firewall engineering, network security operations, and cybersecurity for critical infrastructure. His technical expertise includes Check Point VSX, Citrix ADC/NetScaler, network segmentation, traffic analysis, and secure enterprise network design. Research interests include Zero Trust architecture, micro segmentation, hybrid cloud security, east-west traffic governance, and applied cybersecurity for critical infrastructure.



Harish Apuri is a Software Engineer with experience in building scalable, secure, and high-performance cloud and AI systems across finance, retail, and enterprise domains. He holds a Master's degree in Information Technology and Management from The University of Texas at Dallas and a Bachelor's degree from Jawaharlal Nehru Technological University, Hyderabad. His expertise spans ML Ops, Infrastructure-as-Code, Kubernetes, and multi-cloud platforms, including AWS, Azure, and Google Cloud. He has led the development of production-grade AI platforms that support large-scale machine learning workloads, delivering high availability, reduced costs, and accelerated deployment cycles through automation and CI/CD practices. His research focuses on agentic AI, autonomous cloud operations, and intelligent infrastructure systems. He has authored multiple publications in Springer, IEEE, IJEAT, and Q4-indexed venues, including works on AI-driven root-cause analysis, Auto ML for industrial IoT, multi-agent autonomous workflows, zero-day malware detection, and self-healing infrastructure using LLM agents. He has also contributed to compliance-driven CI/CD security, generative AI for Infrastructure-as-Code resilience, and enterprise AI architecture design. In addition to his research contributions, he has completed 65 peer reviews, served as a publication chair and main editor, participated in hackathon judging, and received a Best Paper Award.



Kishore Bitra, IEEE Senior Member, is an elite enterprise IT architect with over 18 years of experience, specialising in Microsoft 365, Azure Entra identity platforms, and Zero Trust security frameworks. He currently serves as the Lead of Collaboration Engineering for the City of Baltimore's Department of Information Technology. In this capacity, he architects the highly secure, mission-critical cloud infrastructure and digital identity perimeters that support over 13,000 civil servants, ensuring municipal cyber resilience for essential public health initiatives and critical public works utilities. Recognized globally for his technological leadership, Kishore is an IEEE Senior Member and a recently appointed member of the ISACA 2026 Emerging Trends Working Group. He frequently serves as a technical reviewer for international cloud computing and data science conferences and authors thought-leadership articles on enterprise security. Beyond his professional executive roles, Kishore is deeply committed to public safety and community service, serving as a certified first responder and volunteer firefighter with the New Market District Volunteer Fire Company in Maryland, and dedicating his time to digital literacy education with the Literacy Council of Frederick County.

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