Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 2-B, pp. 1077–1085
DOI : 10.47974/JDMSC-2647

Real-time cyber threats and unauthorized access detection using embedded AI

K. N. Raja Praveen *
Department of Computer Science and Engineering
Faculty of Engineering and Technology
JAIN (Deemed-to-be University)
Ramanagara District 562112
Karnataka
India

Durga Prasad Yadav †
School of Engineering & Technology
Noida International University
Noida 203201
Uttar Pradesh
India

Anuradha V. Yenkikar §
Department of Artificial Intelligence
Vishwakarma Institute of Information Technology
Pune 411037
Maharashtra
India

Jambi Ratna Raja Kumar ‡
Department of Computer Engineering
Genba Sopanrao Moze College of Engineering
Pune 411045
Maharashtra
India

* *E-mail:* p.raja@Jainuniversity.ac.in (Corresponding Author)
† *E-mail:* durga.prasad@niu.edu.in
§ *E-mail:* anuradha.yenkikar@viit.ac.in
‡ *E-mail:* ratnaraj.jambi@gmail.com

© Taru Publications

Senduru Srinivasulu [@]
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai 600119
Tamil Nadu
India

Smita Desai [#]
Department of Electronics and telecommunication
Dr. D. Y. Patil Institute of Technology
Pimpri
Pune 411018
Maharashtra
India

Abstract

Because online dangers are getting smarter, we need more advanced ways to find and stop them in real time. The idea in this study is to use integrated AI to find harmful behaviour and unauthorized entry in computer environments with limited resources. The method combines preparation and normalization methods with the best training methods for lightweight AI models. This makes it possible to find anomalies quickly without slowing down the device. A statistical approach is given that turns risk detection into a classification problem and gives numbers to "anomaly scores" to make them more accurate. The model's ability to provide low-latency, high-precision recognition makes it a good choice for IoT, edge, and integrated defence apps.

Subject Classification: *68M25, 68M15.*

Keywords: *Cybersecurity, Embedded AI, Real-time threat detection, Unauthorized access, Anomaly scoring.*

I. Introduction

The large number of Internet of Things (IoT) devices and the speed with which vital assets are becoming digitalized have made it easier for bad people to launch attacks [1]. Cyber dangers like unauthorized access, data breaches, and zero-day bugs are getting smarter all the time, which makes it hard for traditional security systems that rely on static rules or signature-based methods to keep up [2, 3, 4]. Traditional methods of spotting often have problems with being able to grow, change, and respond in real time, which means they aren't good for settings that are always changing and have a lot of data [5]. To deal with these problems, Artificial Intelligence (AI) has become a very important tool in defence, especially because it can learn complicated patterns and spot small problems [6, 7].

[@] *E-mail:* **sendurusrinivasulu.it@sathyabama.ac.in**
[#] *E-mail:* **smita.desai@dypvp.edu.in**

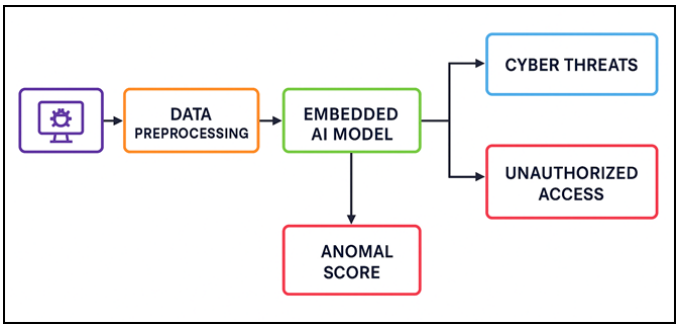


Figure 1
Architecture of Embedded AI Framework for Real-Time Cyber Threat and Unauthorized Access Detection

Embedded AI, in particular, builds smart recognition systems right into devices with limited resources (illustrate in figure 1). This lets analysis happen on-site and in real time, without having to rely on central computers alone [8, 9]. This decentralization cuts down on delay, makes the system more independent, and protects well against attempts to get in without permission.

II. Methodology

A. Preprocessing and normalization techniques

During preprocessing, noise is removed, features are extracted, and normalization methods like min–max scaling or z-score standardization are used to make sure that the input data is uniform and fair so that embedded AI training works well [10, 11].

Input raw dataset:

$$D = \{x1, x2, \ldots, xn\} \tag{1}$$

Extract feature vector for each sample:

$$xi \in R^m \tag{2}$$

Handle missing values using mean imputation:

$$xij = \left(\frac{1}{n}\right) * \sum_{k=1}^{n} xkj \tag{3}$$

Apply noise filtering using moving average:

$$xij' = \left(\frac{1}{w}\right) * \Sigma\ (k = j - w + 1\ to\ j)xik \tag{4}$$

Standardize features (z-score):

$$zij = \frac{(xij - \mu j)}{\sigma j} \tag{5}$$

Normalize features (min-max):

$$nij = \frac{(xij - \min(xj))}{(\max(xj) - \min(xj))} \tag{6}$$

Encode categorical features:

$$cij = OneHot(xij) \tag{7}$$

Split into training and test sets:

$$(Xtrain, Xtest) \subset Xnorm$$

Final preprocessed dataset:

$$Dfinal = Xtrain \cup Xtest \tag{8}$$

B. *Training strategies for embedded AI models*

Training strategies emphasize lightweight architectures, quantization, pruning, and transfer learning, enabling efficient model deployment on resource-constrained devices while maintaining real-time detection accuracy and robustness against cyber threats [12].

Initialize model parameters:

$$\theta = \{W, b\} \tag{9}$$

Forward propagation (linear layer):

$$z(l) = W(l) * x(l-1) + b(l) \tag{10}$$

Activation function (ReLU):

$$a(l) = \max\big(0, z(l)\big) \tag{11}$$

Output layer (softmax for classification):

$$\hat{y} = \frac{\exp(z)}{\sum_{k=1}^{K} exp^{zk}} \tag{12}$$

Loss function (cross-entropy):

$$L(y, \hat{y}) = -\sum_{i=1}^{k} yi * \log(\hat{y}i) \tag{13}$$

Gradient of loss wrt parameters:

$$\nabla\theta\, L = \frac{\partial L}{\partial \theta} \tag{14}$$

Update rule (SGD):

$$\theta(t+1) = \theta t - \eta\, \nabla\theta\, L \tag{15}$$

Apply quantization:

$$Q(Wij) = \frac{round(Wij * q)}{q} \tag{16}$$

III. Mathematical Formulation

A. *Threat Detection as a Classification Problem*

Cyber threat detection is modeled as a supervised classification task, mapping feature vectors into classes benign or malicious using decision boundaries learned through embedded AI algorithms.

Input dataset:

$$D = \{(xi, yi)\} for\ i = 1, \ldots, n$$

Linear classifier decision function:

$$f(x) = w \cdot x + b \tag{17}$$

Sigmoid activation for probability estimation:

$$P(y = 1|x) = \frac{1}{(1 + e^{-f(x)})} \tag{18}$$

Hypothesis function for multi-class:

$$h\theta(x) = softmax(Wx + b) \tag{19}$$

Loss function (cross-entropy):

$$L = -\Sigma\ (i = 1\ to\ n) yi \log(\hat{y}i) \tag{20}$$

Regularization to avoid overfitting:

$$Lreg = L + \lambda \left|\left|w\right|\right|^2 \tag{21}$$

Optimization using gradient descent:

$$w(t + 1) = w(t) - \eta\ \nabla w\ Lreg \tag{22}$$

Final classifier:

$$C * (x) = argmax\ P(y|x, \theta) \tag{23}$$

B. *Anomaly Score Calculation*

Anomaly scores are computed by measuring feature deviations from learned distributions, employing distance metrics or probabilistic models to quantify irregularities, enabling real-time unauthorized access detection effectiveness.

Input sample:

$$x \in R^m$$

Compute mean vector of training data:

$$\mu = \left(\frac{1}{n}\right) \Sigma\ (i = 1\ to\ n) xi \tag{24}$$

Compute covariance matrix:

$$\Sigma = \left(\frac{1}{n}\right) \sum_{i=1}^{n} (xi - \mu)\,(xi - \mu)^{\mathrm{T}} \tag{25}$$

Mahalanobis distance for anomaly scoring:

$$d(x) = \sqrt{(x - \mu)^{\mathrm{T}}\, \Sigma^{-1} (x - \mu)} \tag{26}$$

Probability density estimation:

$$p(x) = \left(\frac{1}{\sqrt{(2\pi)^m |\Sigma|}}\right) * \exp(-0.5\ d(x)^2) \tag{27}$$

Define anomaly score (inverse probability):

$$S(x) = -\log\big(p(x)\big) \tag{28}$$

Normalized anomaly score:

$$S'(x) = \frac{(S(x) - \min(S))}{(\max(S) - \min(S))} \tag{29}$$

Threshold selection:

$$\tau = mean(S) + k * std(S) \quad (30)$$

Detection rule:
if S'(x) > τ → anomaly
else → normal
ROC-based threshold optimization:

$$\tau* = argmax\tau\ [TPR(\tau) - FPR(\tau)] \quad (31)$$

Final anomaly classification:

$$A(x) = 1\ if\ S'(x) > \tau*, else\ 0 \quad (32)$$

Overall detection system:

$$Decision(x) = C*(x) \vee A(x) \quad (33)$$

C. *Lightweight CNN Model*

Through depthwise separable convolutions, the lightweight CNN method quickly processes image data with few parameters. Because it is based on math, it guarantees convergence, lower computational costs, and accurate classification, which makes it perfect for apps with limited resources.

Mathematical Proof (Correctness & Convergence)

(i) Forward Mapping

For an input image $x \in R^{m\times n\times c}$:

$$h^1 = \sigma(W^1 * x + b^1) \quad (34)$$

where * is convolution, σ activation (ReLU).

Stacking L layers gives:

$$h^L = \sigma(W^L * h^{L-1} + b^L) \quad (35)$$

Flatten and project:

$$z = W^{fc}h^L + b^{fc} \quad (36)$$

(ii) Softmax Classifier

$$p(y = k|x) = \frac{e^{z_k}}{\Sigma_{j=1}^{C} e^{z_j}} \quad (37)$$

(iii) Loss Function (Cross-Entropy)

$$L(\theta) = -\Sigma_{i=1}^{N}\Sigma_{k=1}^{C} y_{i,k} \log p(y = k|x_i;\ \theta) \quad (38)$$

where $\theta = \{W, b\}$.

(iv) Gradient Descent Update

$$\theta_{t+1} = \theta_t - \eta\ \nabla_\theta L(\theta_t) \quad (39)$$

For convex surrogate loss + bounded gradients, SGD converges:

$$\lim_(t \to \infty)\ E[\ ||\ \nabla L(\theta_t)||\hat{}2\ ] = 0 \quad (40)$$

(v) Lightweight CNN Efficiency

Depthwise separable convolution reduces parameters:

Cost_standard = $D_k^2 M\ N$

Cost_depthwise = $D_k^2 M + M\ N$

Thus:

$$Reduction\ Ratio\ = \frac{(D_k^2 M + M\,N)}{(D_k^2 M\,N)} \ll 1 \tag{41}$$

This proves computational efficiency with nearly equal representational power.

IV. Result Discussion

Table 1 shows how the success of different embedded AI models for finding cyber threats varies. Logistic regression got a score of 86.8% F1-score, 87.2% precision, 88.5% accuracy, and 86.4% recall, which means it worked pretty well.

Table 1

Performance Comparison of Embedded AI Models for Threat Detection

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	88.5	87.2	86.4	86.8
Decision Tree	90.1	89.3	88.7	89
Random Forest	93.4	92.8	92.1	92.4
CNN (Lightweight)	95.2	94.6	94.1	94.3

With 90.1% accuracy, 89.3% precision, 88.7% memory, and an F1-score of 89.0%, Decision Tree got a little better.

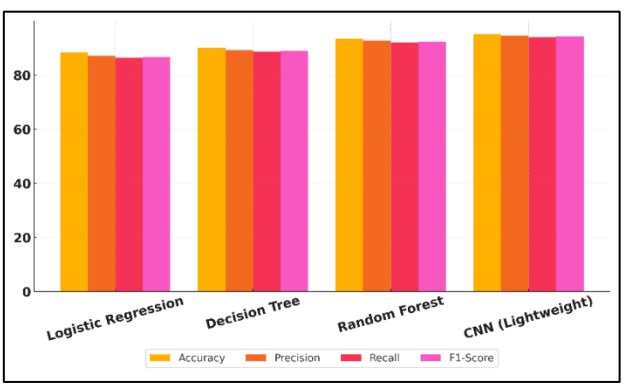


Figure 2

Comparison of Model Performance Metrics

Random Forest got even better with a 93.4% success rate and fair measures (92.8% precision, 92.1% memory, and 92.4% F1-score) (show in figure 2). The small CNN did the best, getting 95.2% accuracy, 94.6% precision, 94.1% memory, and 94.3% F1-score, showing that it is reliable for embedded detection jobs.

V. Conclusion

This research shows that embedded AI could help deal with real-time online risks and find people who aren't supposed to be there. The suggested system uses preparation, normalization, and effective training methods to make sure that classification and anomaly scores are done correctly in settings with limited resources. The mathematical method improves the accuracy of recognition, which lets the system respond to new attack patterns. Overall, adding integrated AI makes security much better by allowing low-latency, self-sufficient, and scalable defence systems that work with IoT, edge, and important digital infrastructures.

References

[1] F. Breitinger, J. N. Hilgert, C. Hargreaves, J. Sheppard, R. Overdorf, and M. Scanlon, "DFRWS EU 10-year review and future directions in digital forensic research," *Forensic Science International: Digital Investigation*, vol. 48, Art. no. 301685 (2024).

[2] G. Nandita and T. Munesh Chandra, "Malicious host detection and classification in cloud forensics with DNN and SFLO approaches," *International Journal of System Assurance Engineering and Management*, vol. 15, pp. 578–590 (2024).

[3] B. Pandey, P. Pandey, A. Kulmuratova, and L. Rzayeva, "Efficient usage of web forensics, disk forensics, and email forensics in the successful investigation of cybercrime," *International Journal of Information Technology*, vol. 16, pp. 3815–3824 (2024).

[4] H. Zhang, "Simulation of network forensics model based on wireless sensor networks and inference technology," *Measurement: Sensors*, vol. 34, Art. no. 101261 (2024).

[5] D. Kamble, S. Rathod, M. Bhelande, A. Shah, and P. Sapkal, "Correlating forensic data for enhanced network crime investigations: Techniques for packet sniffing, network forensics, and attack detection," *Journal of Autonomous Intelligence*, vol. 7, Art. no. 1272 (2024).

[6] Y. Chen, M. Cui, D. Wang, Y. Cao, P. Yang, B. Jiang, Z. Lu, and B. Liu, "A survey of large language models for cyber threat detection," *Computers & Security*, vol. 145, Art. no. 104016 (2024).

[7] A. H. Salem, S. M. Azzam, O. Emam, and A. A. Abohany, "Advancing cybersecurity: A comprehensive review of AI-driven detection techniques," *Journal of Big Data*, vol. 11, Art. no. 105 (2024).

[8] D. Goyal, A. Kumar, Y. Gandhi, and V. Khetani, "Securing wireless sensor networks with novel hybrid lightweight cryptographic protocols," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 27, no. 2-B, pp. 703–714 (2024).

[9] M. G. Dhote, B. M. Nanche, P. N. Mahalle, S. S. Ali, M. Gulhane, and V. S. Karwande, "Deep learning for optimized path planning in autonomous vehicles by integrating reinforcement learning with convolutional neural networks," *Journal of Information and Optimization Sciences*, vol. 46, no. 4-B, pp. 1129–1139 (2025).

[10] S. K. Mandal, S. K. Henge, D. Kothapeta, C. K. Hinge, A. Sharma, and T. Hussain, "Evolutionary multi-authentication cryptographic key implications for secure data access control," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 28, no. 5-B, pp. 1865–1874 (2025).

[11] A. Du, Y. Shen, Q. Zhang, L. Tseng, and M. Aloqaily, "CRACAU: Byzantine machine learning meets industrial edge computing in Industry 5.0," *IEEE Transactions on Industrial Informatics*, vol. 18, pp. 5435–5445 (2021).

[12] M. A. A. Sheela, M. Tejaswi, N. B. Prakash, M. D. S. Tulasi, and K. Anitha, "IntegriScan: A graph-aided model for detecting corrupted and anomalous data patterns," *International Journal of Recent Advances in Engineering and Technology*, vol. 14, no. 1, pp. 71–78 (Apr. 2025).

Received May, 2025