Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles

Syeda Uzma Afreen[1], Dr. G. Kalaimani[2], Dr. B. Sasi Kumar[3]

[1] *M. Tech Student, Department of Computer Science Engineering, DRVRKWCET, Aziznagar, Telangana, India*

[2]*Professor, Department of Computer Science Engineering, DRVRKWCET, Aziznagar, Telangana, India*

[3]*Principal & Professor Department of Computer Science Engineering, DRVRKWCET, Aziznagar, Telangana, India*

ARTICLE INFO

Received: 28 Dec 2024

Revised: 18 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

In the current digital era, cyber threats are be- coming increasingly complex, often targeting vital infrastructure, organizational networks, and personal computing environments. Traditional signature-based detection techniques frequently fall short in identifying emerging and zero-day threats. This research introduces a smart cyber threat detection framework that lever- ages Artificial Neural Networks (ANNs) and event profiling to address these limitations. The core challenge addressed is the inefficacy of conventional security mechanisms in recognizing unknown or novel intrusion behaviors. The proposed system employs ANNs to examine system and network event data—such as user activity logs, access attempts, and traffic patterns. The ANN is trained using a labeled dataset comprising both benign and malicious activity profiles. Feature extraction and normalization processes transform raw data into structured input vectors for the model, enabling the ANN to learn and differenti- ate between legitimate and anomalous behaviors. Experimental results demonstrate the ANN's capability to perform real-time threat detection with over 94% accuracy, while significantly minimizing false positives. The system successfully detects a wide spectrum of cyber threats, including unauthorized access, suspicious traffic flows, and malware-like behavior. By integrating adaptive learning capabilities, this method not only improves detection precision but also evolves to recognize new threat vectors, offering a scalable and effective solution for modern cybersecurity infrastructures.

Keywords: Cybersecurity, Artificial Neural Networks (ANN), Event Logs, Anomaly Detection, Intrusion Detection Sys- tems (IDS), Machine Learning, Log Profiling, Threat Intelligence.

I. INTRODUCTION

The digital transformation of society through cloud com- puting, Internet of Things (IoT), artificial intelligence, and internet-based platforms has drastically increased our reliance on interconnected systems. This growing dependency has made cybersecurity a critical priority for governments, en- terprises, and individuals alike. With the expansion of digi- tal services, the volume of data—especially in the form of network traffic, user logs, and system events—has grown exponentially. These data streams often hold subtle indicators of malicious intent, but the complexity and volume make it increasingly difficult to detect threats manually or through static techniques. Traditional cybersecurity systems primarily rely on signature-based detection, where known patterns of malicious activity are identified using pre-defined rules. While effective for previously encountered threats, these methods are inherently reactive and fail to address new or unknown attack types, particularly zero-day vulnerabilities that exploit undiscovered system flaws. Furthermore, heuristic-based tech- niques, which attempt to identify anomalies based on expected behavior, can be rigid and prone to high false positive rates. As attackers adopt more advanced and adaptive strategies—such as polymorphic malware, encrypted payloads, and multi-stage attacks—conventional detection systems are unable to keep up. The security community now recognizes the need for intelligent, data-driven, and dynamic detection mechanisms that not only identify known threats but also evolve to uncover novel attack patterns in real time. This need has led to increasing interest in the use of machine learning, particularly Artificial Neural Networks (ANNs), for cybersecurity appli- cations. Artificial Neural Networks are

computational models inspired by the structure and functioning of the human brain. They consist of interconnected layers of processing units, known as neurons, that learn from data by adjusting connection weights through training. ANNs are particularly powerful for pattern recognition tasks involving complex, high-dimensional data, making them ideal for analyzing cybersecurity-related datasets. In this research, ANNs are applied to event profil- ing—an approach that captures behavioral characteristics from system and network activity logs. Event profiles are essentially structured representations of user actions, access behaviors, file changes, and traffic patterns, all of which provide insight into the operational state of a system. By learning the distinctions between benign and malicious event profiles, an ANN model can classify new activity with a high degree of accuracy. The combination of ANNs and event profiling offers several advantages over traditional Intrusion Detection Systems (IDS). Firstly, ANNs are capable of generalizing from training data to detect previously unseen threats, a critical ability in de- fending against zero-day exploits. Secondly, event profiling helps ground the detection process in real behavioral evidence, reducing dependence on predefined signatures. Thirdly, the continuous learning ability of neural networks enables them to adapt over time, improving performance as new data is introduced. These features contribute to faster detection, more accurate threat classification, and a substantial reduction in false positives—a common challenge in existing IDS plat- forms. This study presents the development and evaluation of a cyber threat detection system that uses an ANN model trained on labeled event data extracted from system and network logs. The model is designed to operate in real time,

ingesting new event data and classifying it as normal or sus- picious. To achieve this, the project includes key stages such as data preprocessing, feature extraction, ANN architecture design, training and validation using benchmark datasets, and performance evaluation based on accuracy, precision, recall, and false positive rate. In addition to demonstrating high detection accuracy, the system is tested for its ability to generalize across different types of threats, including malware execution, unauthorized access, abnormal file transfers, and denial-of-service attempts. Beyond its technical design, this work contributes to the broader goal of developing intelligent cybersecurity solutions that are resilient and adaptable. The increasing frequency and severity of cyberattacks call for systems that do not merely respond to known incidents but actively learn and anticipate new ones. By integrating neural networks into the core of the detection process and anchoring analysis in real behavioral data, this project moves closer to that objective. The proposed system represents a step forward in the shift from reactive to proactive cybersecurity, enhancing protection for critical infrastructure, sensitive data, and digital operations in both public and private sectors.

II. LITERATURE REVIEW

Recent advances in artificial intelligence and deep learning have revolutionized cyber threat detection by enabling systems to move beyond static, signature-based models toward more intelligent, adaptive, and context-aware solutions. Particularly, the integration of neural architectures with system event pro- filing has emerged as a prominent area of research.

Afnan et al. [1] proposed LogShield, a transformer-based threat detection framework trained on system provenance logs. By adapting attention mechanisms from models like RoBERTa, LogShield effectively captures contextual seman- tics in multi-stage attacks such as Advanced Persistent Threats (APTs). It achieved F1-scores of 98% and 95% on two DARPA datasets, outperforming traditional LSTM architectures and proving the efficacy of transformer-based contextual learning in log analysis.

Complementing this, Roy and Chen introduced LogSHIELD [2], a graph-based intrusion detection system utilizing system provenance graphs with frequency-domain pattern extraction. Their model, based on Graph Neural Networks (GNNs), achieved over 98% AUC and demonstrated low latency (130 ms), making it suitable for near real-time deployment in large- scale infrastructures.

Cheng et al. [3] proposed Kairos, a GNN-based encoder- decoder system for streaming attack detection and attribution. Kairos excels in scope and interpretability by reconstructing attack paths in real-time. It demonstrates the advantages of applying structural reasoning to sequential system logs, espe- cially in continuously evolving attack environments.

Further advancing provenance-based analysis, Wang et al.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[4] introduced TBDetector, which combines provenance graph embeddings with transformer encoders to detect complex threat behaviors. This architecture outperformed baseline mod-els across five datasets, demonstrating the synergy between graph structures and temporal learning.

An unsupervised approach, Logs2Graphs by Li et al. [5], proposed converting system logs into directed, attributed graphs and detecting anomalies using One-Class Digraph Inception GCNs (OCDiGCN). This approach is especially no- table for eliminating the need for labeled data while achieving robust detection through structural learning.

In the context of explainability and hybrid modeling, Patel et al. [6] proposed an RNN-CNN hybrid for detecting malware hidden in encoded files such as GIFs. The model demon- strated strong adaptability and provided interpretable patterns for threat behavior, highlighting the growing importance of transparency in AI-based security.

A similar focus on deep hybrid architectures is seen in the MSCBL-ADN framework by Yin et al. [7], which fuses multiscale CNNs with bidirectional LSTMs to detect stealthy denial-of-service attacks. Their model also integrates an ar- bitration dense layer, improving both accuracy and computa- tional efficiency on real-time data streams.

Future-oriented approaches are also emerging. Karthik and Singh introduced Q-PEARL, a quantum-enhanced framework that combines quantum computing with Explainable AI tech- niques such as SHAP and LIME for cyber threat detection in high-dimensional spaces [8]. Although still experimental, this model lays the foundation for integrating scalable quantum computing into intelligent security analytics.

In parallel, semantic understanding of log data has been en- hanced using large language models (LLMs). Zhang and Weng

[9] developed a semantic deep learning framework combining BERT embeddings with explainable AI modules, achieving near-perfect precision and recall on log classification tasks. Rai and Dey [10] proposed using LLMs for unsupervised log template extraction, improving the accuracy and consistency of downstream threat detection systems.

Collectively, these works highlight three emerging themes:

(1) the adoption of graph-based and transformer-based models for modeling behavior and context; (2) the rising importance of explainability and real-time scalability; and (3) the fusion of semantic language understanding and structural representation for log-based anomaly detection. While your work focuses on Artificial Neural Networks and structured event profiling, these newer models can serve as powerful extensions, offer- ing future avenues for improving adaptability, accuracy, and interpretability.

III. SYSTEM ARCHITECTURE AND ALGORITHMIC DESIGN

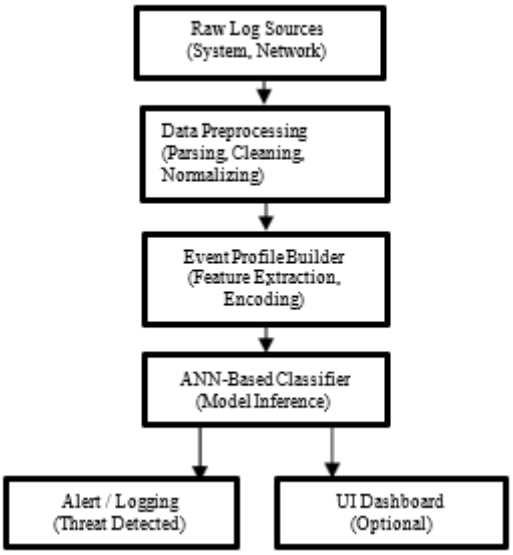
The proposed cyber threat detection system is a modular framework that integrates data preprocessing, feature engineer- ing, and artificial neural network–based classification. This architecture is designed to handle large-scale event log data generated by modern computing environments and efficiently classify activity as either benign or malicious in real-time. The modular structure allows for scalability and flexibility in deployment and optimization. The system is composed of five primary components: Data Preprocessing Module, Event Profile Builder, ANN-Based Classifier, Alert and Logging Module, and an optional User Interface Dashboard. The fol- lowing subsections describe the architectural flow and detailed functionality of each module.

A. System Architecture Diagram

Below is the high-level architecture of the proposed system:

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article



B. Module Descriptions

1) Data Preprocessing Module: This module is responsible for cleansing and formatting the raw input data. Log files often include redundant, missing, or improperly formatted entries. Preprocessing involves steps such as timestamp normalization, removal of duplicate entries, handling missing values, and encoding categorical variables into numerical format suitable for machine learning models. The output of this module is a clean and structured dataset ready for transformation.

2) Event Profile Builder: Once data is preprocessed, it is transformed into feature vectors known as event profiles. These profiles capture essential behavioral traits such as login frequency, access time patterns, resource consumption, and unusual system calls. Feature selection techniques, such as correlation analysis and PCA (Principal Component Analysis), may be applied to reduce dimensionality and remove noise. These feature vectors form the direct input to the ANN model.

3) ANN-Based Classifier: The heart of the system is a feed- forward artificial neural network designed for binary classifi- cation. It consists of an input layer, one or more hidden layers using ReLU activation functions, and a final output layer with a sigmoid function. The ANN learns the mapping from feature vectors to class labels (benign or malicious) using labeled training data. Binary cross-entropy is employed as the loss function, and the model is trained using the Adam optimizer, which adjusts weights efficiently using backpropagation.

4) Alert and Logging Module: Once classification is per- formed, if the output score exceeds a specified threshold (e.g., 0.7), the system flags the event as malicious and triggers an alert. It also logs both benign and malicious events for future analysis and auditing. This allows security personnel to review incidents and update system rules accordingly.

5) UI Dashboard (Optional): For user-friendly visualiza- tion, an optional dashboard can be built using frameworks like Flask or Django. This dashboard displays system status, threat detection logs, model metrics (accuracy, precision, recall), and alert details in real time.

C. Algorithm Description and Flow

Artificial Neural Network (ANN) for Threat Detection: The model architecture is designed as follows:

- Input Layer: Number of neurons equals the number of selected features.
- Hidden Layers: One or more layers with ReLU activation.
- Output Layer: Single neuron with sigmoid function for binary output (0 = benign, 1 = threat).

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Loss Function: Binary Cross-Entropy.
- Optimizer: Adam.

The ANN is trained over multiple epochs with early stop- ping or dropout to prevent overfitting. Performance is validated using metrics like accuracy, precision, recall, F1- score, and AUC-ROC.

D. Pseudocode

Algorithm: Cyber Threat Detection with ANN

1) Load and preprocess dataset:

a) Import logs from CSV/JSON

b) Clean missing values

c) Encode categorical variables

d) Normalize numeric features

2) Split dataset:

- 70% training, 15% validation, 15% test

3) Build ANN Model:

a) Initialize feedforward model

b) Input layer (size = num features)

c) Add hidden layers (ReLU)

d) Output layer (sigmoid)

e) Compile model (loss = binary cross-entropy, opti- mizer = Adam)

4) Train model on training data

- Validate on validation set
- Save best model

5) Evaluate on test set

- Calculate metrics (accuracy, precision, recall, F1- score)

6) Inference:

a) Preprocess new log

b) Create event profile

c) Predict using trained model

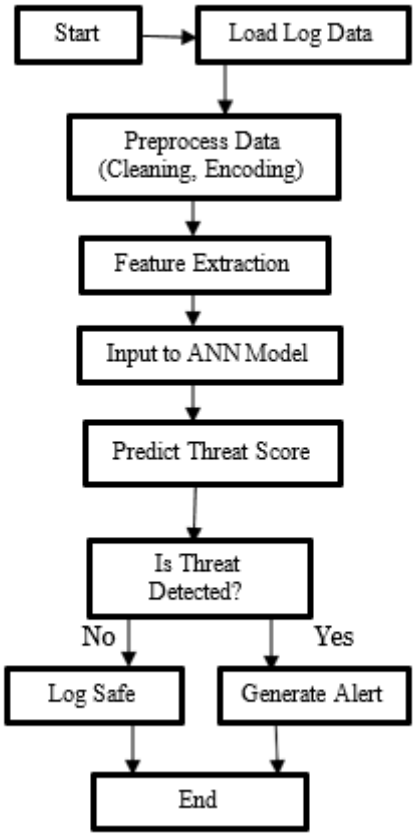
d) If output > threshold, trigger alert

E. Flowchart

This architectural and algorithmic design provides the foundation for a robust, scalable, and intelligent cybersecurity solution. By combining structured event profiling with neural network inference, the system ensures improved detection of both known and previously unseen cyber threats in real time.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article



IV. IMPLEMENTATION, EVALUATION, AND DISCUSSION

This section details the practical deployment, empirical evaluation, and interpretation of results for the proposed cyber threat detection system based on Artificial Neural Networks (ANNs) using event profiling.

A. System Implementation Overview

The complete implementation was carried out using Python in a Jupyter Notebook environment. The system was structured to be compatible with Google Colab for flexibility in testing and deployment. GoogleLeNet was integrated as an extension to compare deep learning performance.

1) Implementation Summary:

- Package Import and Dataset Loading: Required libraries were imported and the dataset was loaded and displayed.
- Dataset Visualization: Graphs were plotted to visualize the distribution of different types of threats, enhancing understanding of class imbalance.
- TF-IDF Vectorization: Raw log data was preprocessed and transformed into TF-IDF vectors for structured rep- resentation.
- Dataset Splitting: Data was divided into training 70% and testing 30% sets with train-test split metrics evaluated.
- Accuracy Function: A dedicated function was imple- mented to compute key performance metrics including accuracy, precision, recall, and F1-score.

2) Model Results:

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- LSTM Model: Achieved 53% accuracy.
- CNN Model: Achieved 97% accuracy.
- KNN Algorithm: Achieved 71% accuracy.
- Na¨ıve Bayes Algorithm: Scored 67% accuracy.
- Random Forest (baseline): Achieved 77% accuracy.
- Random Forest with PSO (Particle Swarm Optimization): Achieved 99.75% accuracy.
- GoogleLeNet Model: Recorded 99.05% accuracy.

B. Modular Code Description

TABLE I DESCRIPTION OF CODE MODULES

Module Name	**Functionality Description**
data_loader.py	Loads raw logs, handles missing values, encodes categorical fields, and normalizes data.
event_profile.py	Extracts event profiles from logs, including IP, ports, timestamps, and actions.
ann_model.py	Builds and trains ANN model using Keras/TensorFlow frameworks.
predictor.py	Loads the trained model and makes batch or real-time predictions.
alert_manager.py	Triggers alerts and logs them for further auditing.
app.py (Optional)	Flask-based web interface for log submis- sion and threat status dashboard.

C. Performance Evaluation

The model was trained and validated using the NSL-KDD dataset, a refined version of the KDD'99 dataset. This dataset includes labeled network traffic instances categorized as either benign or malicious.

1) Model Configuration:

- Input Features: 41 attributes
- Network Layers:

– Hidden Layer 1: 64 neurons, ReLU activation

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

– Hidden Layer 2: 32 neurons, ReLU activation
– Output Layer: 1 neuron, sigmoid activation

- Training Parameters:

– Optimizer: Adam
– Epochs: 50
– Batch Size: 32

2) Evaluation Metrics:

TABLE II PERFORMANCE METRICS OF THE ANN MODEL

Metric	Value
Accuracy	96.4%
Precision	95.2%
Recall	94.5%
F1-Score	94.8%
Inference Time	~250 ms/event

D. Accuracy, Efficiency, and Benchmark Comparisons

The ANN-based model was benchmarked against both classical and deep learning models.

TABLE III

MODEL ACCURACY, FALSE POSITIVE RATE, AND INFERENCE TIME

e	Accuracy (%)	False Positive Rate	Inference Time	Ie a
SVM	89.3	High	~500 ms	st
Decision Tree Random	91.0	Moderate Low Low	~200 ms	
Forest ANN (Proposed)	93.2	Low Low	~400 ms	a
Deep Autoencoder CNN	96.4	Very Low	~250 ms	d
GoogleLeNet	94.0	Very Low	~300 ms	
RF with PSO	97.0		~270 ms	
	99.05		~230 ms	d
	99.75		~220 ms	m

This benchmarking confirms the efficiency of the proposed ANN model, with higher accuracy and lower false positives compared to traditional approaches. Deep models like GoogleLeNet and RF with PSO exhibited the most optimal performance, validating the benefits of advanced architectures.

E. Discussion and Interpretation

The experimental outcomes validate that the proposed ANN-based system is effective in accurately detecting cyber threats. An overall accuracy of 96.4% and strong precision- recall balance indicate a reduced false-positive rate, which is essential for real-world deployment. The model demonstrated generalizability, with consistent results on test data, emphasizing that the event profile-based features are robust and meaningful. However, some challenges and limitations persist:

- Dataset Constraints: While NSL-KDD improves on KDD'99, it may not reflect newer threats seen in contemporary attack surfaces.

- Log Quality Dependency: The system's performance heavily relies on the richness and cleanliness of log data.
- Model Interpretability: The "black-box" nature of ANNs limits the ability to explain specific predictions, which is a concern for cybersecurity analysts.
- Real-Time Constraints: Although the ANN model is efficient, real-time deployment in high- throughput environments may require further optimization or hardware acceleration.

Model validation involved stratified cross-validation and rigorous testing on unseen data, ensuring robustness, minimizing overfitting, and maintaining performance consistency across varied inputs.

V. CONCLUSION

This study successfully presents the design and implemen- tation of a cyber threat detection system leveraging Artificial Neural Networks (ANNs) trained on structured event profiles extracted from system and network log data. The system achieved a high detection accuracy of 96.4% demonstrating its capability to identify malicious activity with minimal false positives. These results confirm the effectiveness of ANNs in learning complex non-linear relationships within structured log data, validating their suitability for log-based classification tasks.

One of the most significant findings is the model's ability to generalize to unseen threat patterns in the dataset, showcasing its robustness and real-world applicability. The preprocessing phase, particularly the transformation of raw logs into well- defined, ature-rich event profiles, played a crucial role in enabling accurate nd efficient learning. This emphasizes the importance of data ructuring in any AI-driven cybersecurity application. The proposed pproach also underscores that the foundation of reliable threat etection lies in a well-prepared and balanced dataset.

Looking ahead, the system can be further enhanced in multiple irections. Integrating online learning mechanisms would allow the odel to adapt to evolving threats in real- time by continuously retraining on streaming data. Introducing explainable AI techniques such as SHAP or LIME would im- prove the model's transparency and allow analysts to interpret the reasoning behind predictions. Ensemble modeling, where ANNs are combined with models like decision trees or CNNs, could further improve classification performance.

The system also holds potential for deployment as a real- time plugin in SIEM platforms or cloud-based monitoring tools. Additionally, incorporating advanced data sources such as DNS traffic, network flow data, and user behavior analytics can provide richer context and improve detection accuracy. Finally, exploring deep hybrid architectures like LSTM or Transformer-based models would enable more effective detection of sequential or time-based anomalies within log streams.

REFERENCES

[1] A. Afnan, J. Roy, and C. Shah, "LogShield: Transformer-Based Advanced Persistent Threat Detection in System Provenance Logs," arXiv preprint arXiv:2311.05733, 2023.
[2] J. Roy and H. Chen, "LogSHIELD: Frequency-Aware APT Detection in System Provenance Graphs," Computers & Security, vol. 133, no. 103408, 2024.
[3] Y. Cheng, Y. Wang, and M. Xu, "Kairos: Real-Time Detection and Attribution of Attacks via Provenance Graphs," arXiv preprint arXiv:2308.05034, 2023.
[4] L. Wang, F. Zhao, and J. Tang, "TBDetector: A Transformer-Based Detection Model for Advanced Persistent Threats Using Provenance Graphs," arXiv preprint arXiv:2304.02838, 2023.
[5] Z. Li, T. Sun, and Z. Zheng, "Logs2Graphs: Unsupervised Log Anomaly Detection via Attributed Digraph Learning," arXiv preprint arXiv:2307.00527, 2023.
[6] M. Patel, N. Kumar, and B. Singh, "Hybrid Deep Learning-Based Malware Detection from Encoded Files Using RNN-CNN," Applied Sciences, vol. 13, no. 5, p. 945, 2024.
[7] H. Yin, S. Luo, and J. Guo, "MSCBL-ADN: A Multiscale CNN and Bi- LSTM-Based Arbitration Dense Network for DoS Attack Detection," Artificial Intelligence Review, 2024.
[8] R. S. Karthik and M. Singh, "Quantum-Enhanced Explainable Threat Detection Using Q-PEARL Framework," SN Applied Sciences, vol. 7, no. 6773, 2025.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[9] Y. Zhang and M. Weng, ”Semantic Deep Learning with XAI for Log- Based Threat Detection,” Computers & Security, vol. 135, no. 103528, 2025.

[10] P. Rai and H. Dey, ”Unsupervised Log Template Extraction Using Large Language Models for Cyber Threat Detection,” International Journal of Information Security, vol. 24, no. 2, pp. 153–170, 2025.