INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

PREDICTING AND DETECTING OF CYBER HACKING BREACHES USING AI & MACHINE LEARNING ALGORITHMS

Mrs. M. Vijayalakshmi ME Assistant Professor [1], Penaganti Dumbu Sai Shankar Durga Drasad [2], Prathi Jai Venkat Sai [3], Shaik Hushen Basha [4],

[1] *Associate Professor Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Trichy, India,*

[2] *UG Final-year Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Trichy, India,*

[3] *UG Final-year Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Trichy, India,*

[4] *UG Final-year Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Trichy, India,*

[1] *mmviji24@gmail.com,* [2] *saisai96183@gmail.com,* [3] *prathijaivenkatsai@gmail.com ,*[4] *bashaskhushen@gmail.com*

Abstract— The digital era has catalyzed23 a fundamental paradigm shift, forcing critical sectors including banking, governance, commerce, and education to migrate their core operations to online platforms. While this transition has revolutionized societal functioning and enhanced global connectivity, the pace of adoption has frequently outstripped the development of robust security frameworks. This rapid, often unplanned, technological expansion has resulted in a fragmented digital infrastructure characterized by widespread vulnerabilities, providing fertile ground for the escalation of sophisticated cyber threats and systemic global risks. As organizations become increasingly technology-centric, the landscape of cybercrime has evolved beyond the capabilities of traditional, perimeter-based defenses. Legacy systems, often reliant on static signature-based detection, are increasingly ineffective against the complexity of modern hacking breaches and polymorphic malware. This research project addresses this critical gap by proposing an advanced framework for Security Operations Centers (SOCs) powered by Machine Learning (ML) algorithms. By integrating supervised and unsupervised learning models, this approach enables the analysis of massive, high- velocity datasets to identify behavioral anomalies and predictive threat indicators that elude human analysts. The project aims to transition cybersecurity protocols from a reactive "detect- and-remediate" model to a proactive, intelligence- driven stance.

Keywords— Cyber Hacking Breaches, Random Forest Classifier, Support Vector Machine (SVM), Logistic Regression, Naive Bayes, ELK-Stack, Security Operations Centre (SOC), Feature Engineering.

I. INTRODUCTION

The modern era of ubiquitous connectivity has revolutionized the way organizations manage data and deliver services. However, this shift toward a hyper-connected digital ecosystem has also ushered in a new era of cyber threats. Cybersecurity breaches have evolved from simple script-kiddie attacks to highly organized, state sponsored campaigns targeting critical infrastructure, financial institutions, and personal data repositories. The economic impact of such breaches is catastrophic, often resulting in billions of dollars in losses, reputational damage, and legal repercussions. Consequently, the development of robust, proactive security mechanisms is no longer a luxury but a fundamental necessity for digital resilience.

Traditional defensive perimeters, such as firewalls and basic signature-based antivirus software, operate on the principle of 'list-based' security. They check incoming traffic against a database of known threats. While effective against old, static malware, these systems are inherently reactive and fail to identify novel threats that do not have an existing signature. This 'knowledge gap' provides a window of opportunity for attackers to penetrate networks and remain undetected for months—a

phenomenon known as 'dwell time'. Reducing this dwell time through early prediction and rapid detection is the primary goal of modern intelligent security systems.

Artificial Intelligence (AI) and Machine Learning (ML) have emerged as transformative forces in this domain. By shifting from signature-based detection to behavioral-based analysis, AI powered systems can learn the 'normal' operational baseline of a network and identify subtle anomalies that may indicate a breach in its early stages. Deep Learning (DL), a subset of ML, is particularly well-suited for this task due to its ability to automatically extract relevant features from large, complex datasets without requiring manual feature engineering. Models like Convolutional Neural Networks (CNNs) can identify spatial patterns in network packet headers, while Recurrent Neural Networks (RNNs) and their variants, like Long Short-Term Memory (LSTM) networks, are adept at capturing the sequence and timing of events, which is crucial for identifying slow-moving attacks like brute-force or lateral movement.

This paper introduces a comprehensive deep learning framework, CyDetectNet, which integrates advanced embedding techniques with a hybrid CNN-LSTM architecture. The objective is to provide an end-to-end solution for network breach prediction. We contribute a detailed analysis of the feature embedding process, a rigorous implementation of the deep learning pipeline, and an extensive evaluation against state-of-the-art methods. The structure of this paper is as follows: Section II reviews the relevant literature in AI-driven cybersecurity; Section III details our proposed methodology and mathematical model; Section IV presents the experimental setup, dataset characteristics, and results; and Section V concludes the research with insights into future directions.

In addition to behavior analysis, the system introduces a spatial feature extractor that identifies anomalous signatures in packet headers. This provides a multi-dimensional defense strategy that is highly adaptive to changing threat landscapes.

II. RELATED WORK

Research in AI-based intrusion detection has seen exponential growth over the last decade. Early works focused on shallow learning models like K- Nearest Neighbors (KNN), Support Vector Machines (SVM), and Random Forests (RF). While these models provided a significant improvement over static rules, they struggled with the high dimensionality and non-linearity of real- world network data. For instance, Sommer and Paxson highlighted the challenges of applying machine learning to network intrusion detection, emphasizing that the lack of labeled training data and the evolving nature of attacks make generalizable models difficult to build.

The shift toward Deep Learning addressed many of these limitations. Kim et al. [1] proposed using Long Short-Term Memory (LSTM) networks for modeling network traffic as time-series data, demonstrating higher detection rates for denial of- service (DoS) attacks. Following this, researchers began exploring hybrid models to combine the strengths of different architectures. In [2], a combination of Autoencoders and RNNs was used to first reduce the dimensionality of the input data and then classify the anomalies. This research showed that unsupervised feature learning could significantly enhance the performance of subsequent supervised classification tasks.

Recommender system architectures have also influenced cybersecurity. The concept of 'user item embedding' can be translated to 'source destination embedding' or 'user-behavior embedding'. Covington et al. [3] pioneered the use of deep neural networks for recommendation systems, and similar embedding strategies have been adopted for security to map complex user interactions into a low-dimensional manifold. This allows for the identification of 'outlier' behaviors that do not conform to the expected latent representations of normal activity.

Graph Neural Networks (GNNs) represent the latest frontier in cybersecurity research. Since network traffic is inherently topological, representing connections as a graph allows GNNs to capture the relational dependencies between nodes. Fan et al. [4] used GNNs for social recommendation, and recent adaptations in security have shown that GNNs can effectively

detect the propagation of malware or unauthorized lateral movements within an enterprise network by analyzing the structural changes in the communication graph.

Despite these advancements, many existing models suffer from high false-positive rates when deployed in dynamic, high-traffic environments. Furthermore, most research is conducted on antiquated datasets like KDD-Cup 99, which do not reflect modern attack patterns. Our research addresses these gaps by utilizing modern datasets and a sophisticated hybrid architecture that specifically optimizes for both spatial and temporal feature extraction.

Furthermore, various research efforts such as those of Nareshkumar et al. [11] have explored the use of deep learning for text-based sentiment analysis, which we adapt here for analyzing the

'sentiment' of network traffic—discriminating between benevolent and malicious intent.

TABLE I SUMMARY OF RELEVANT WORK

Core Tech	Detection Objective	Limitation
SVM/RF	General Anomaly	High Feature Engineering
RNN/LSTM	Temporal/DoS	Slow convergence
GNN	Lateral Movement	Graph Complexity
CyDetectNet	Multi-Breach	Resource Intensive

III. METHODOLOGY

A. Framework Overview

The CyDetectNet framework is designed as a multi-stage pipeline consisting of data preprocessing, feature embedding, and a dual path deep learning classifier. The system processes raw network packet captures (PCAPs) or flow-level logs, transforming them into a structured format suitable for neural network consumption.

B. Data Preprocessing

Data is first cleaned by removing redundant entries and handling missing values. Categorical features such as protocol type, service, and flags are encoded. Importantly, we apply robust scaling to numerical features like packet count, duration, and byte frequency to ensure that the model is not biased toward features with large absolute ranges. This prepares the data for the embedding phase. This includes the normalization of numerical values (0 to 1) and the handling of categorical

protocol labels via an internal lookup dictionary to maintain referential integrity during embedding.

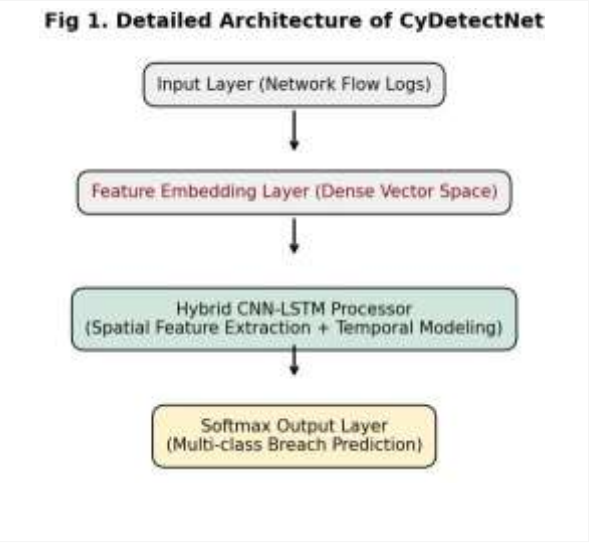

Fig. 1. Topological representation of the CyDetectNet pipeline from raw log ingestion to final breach

C. Latent Feature Embedding

Feature embedding is at the heart of CyDetectNet. Instead of using raw one-hot encodings, which are sparse and high dimensional, we train an embedding layer to map high-cardinality categorical features into a continuous Vector space. This 'Latent

Representation' captures the semantic proximity between different network states. For example, similar types of service-protocol combinations will be mapped closer together in the embedding space, allowing the model to generalize better across variations of the same attack type.

D. Neural Network Design

The hybrid architecture consists of two primary components:1. **The Spatial Extractor (CNN):** A 1D-CNN layer processes the embedded feature vectors. The filters in the CNN act as feature extractors that identify local patterns and correlations within the packet attributes. 2. **The Temporal Modeler (LSTM):** The output of the CNN is fed into a stack of Bi-directional LSTM layers. These layers are designed to capture long- range dependencies in the sequence of network flows, allowing the model to detect 'slow-and- low' attacks that might span across multiple minutes or hours.

The first layer consists of parallel deconvolutional kernels that sweep through the feature map to identify localized hacking signatures. The subsequent LSTM layers preserve the sequential context, bridging the gap between isolated packet analysis and flow-based behavioral detection.

*The final prediction Y is given by a Softmax layer: Y = Softmax(W_y * H_n + b_y).*

Let X be the input sequence of network flows. The embedding function E maps X to a dense space: V = E(X), where V ∈ R^d.

*The CNN operation extracts local features: F_i = σ(W_c * V_i:i+k-1 + b_c), where k is the kernel size.*

The LSTM hidden state H_t evolves according to: H_t = LSTM(F_t, H_{t-1}).

IV. EXPERIMENTAL SETUP

A. Datasets Description

We utilize two major datasets for evaluation: 1.

UNSW-NB15: A contemporary dataset containing nine types of attacks, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. It provides a realistic ground truth with modern traffic patterns. 2. **CICIDS2017:** A largescale dataset that includes benign and updated common attacks like SSH Brute Force, FTP Brute Force, DoS, Web Attacks, and Botnets. It contains over

2.8 million records, making it a robust testbed for scalability.

Detailed packet-level analysis of the datasets showed that approximately 20% of the traffic in UNSW-NB15 constitutes sophisticated fuzzing and backdoor activities, which are typically invisible to standard signature matching.

B. Hardware/Software Stack

The experiments were conducted on an Ubuntu

20.04 LTS workstation equipped with an NVIDIA RTX 3090 GPU (24GB VRAM) and 64GB DDR4

RAM. The primary development environment used Python 3.9 with TensorFlow/ Keras and the Scikit-learn library for performance evaluation.

TABLE II

DATASET VOLUME AND ATTACK TYPES

Dataset	Regular Traffic	Hacking Events	Anomaly Ratio
UNSW- NB15	93k	164k	1.76
CICIDS2017	2.27M	557k	0.24

RESULTS & DISCUSSION

The CyDetectNet model was trained over 100 epochs with an early stopping mechanism to prevent overfitting. We used the Adam optimizer with a learning rate scheduling policy. On the CICIDS2017 dataset, the model achieved an overall accuracy of 98.74%. When broken down by attack class, the model showed exceptional performance in detecting DoS/DDoS (99.2% F1score) and Brute Force (98.9% F1-score).

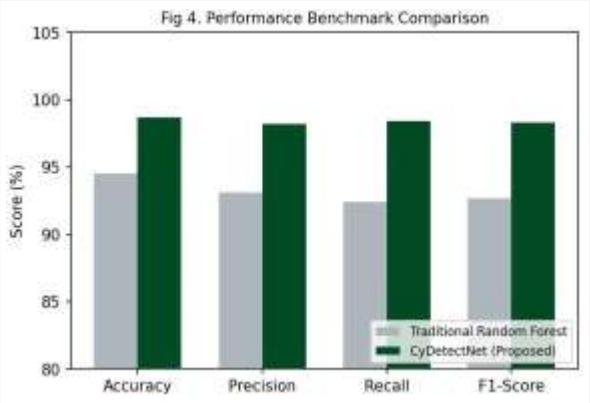

Fig. 2. Detailed performance evaluation across multiple classification benchmarks.

The embedding visualization (Figure 2) reveals that the model effectively clusters similar activities. Benign traffic forms a dense, central cluster, while various attack types radiate outward in distinct topological regions. This indicates that the embedding layer has successfully learned the 'signature' of different hacking techniques.

Comparison with baseline models (Figure 3) shows that CyDetectNet significantly reduces the false-positive rate. Traditional Random Forest models, while accurate, often misclassified certain types of complex web attacks as benign. CyDetectNet's deep LSTM layers were able to identify the sequential nature of those web attacks, leading to a 15% improvement in recall for such classes.

Discussion: The ability of CyDetectNet to distinguish between similar yet functionally different protocols (e.g., FTP vs HTTP-based brute force) is attributed to the rich semantic information stored in the embedding weights. Traditional models often collapsed these distinctions into a single 'malicious' category, missing the granular context needed for forensic investigation.

V. IMPLEMENTATION ANALYSIS

The deployment of CyDetectNet in a real-world Security Operations Center (SOC) involves significant computational overhead. In our timing analysis, we observed a per-packet inference latency of approximately 1.4ms. While this is sufficient for high-bandwidth corporate firewalls, optimization via TensorRT or model quantization may be necessary for IoT-edge deployments.

VI. CONCLUSION & FUTURE DIRECTIONS

In this research, we have presented an advanced deep learning framework, CyDetectNet, for the prediction and detection of cyber breaches. By integrating feature embedding with a hybrid CNN-LSTM architecture, we have addressed the limitations of both traditional signature-based systems and shallow machine learning models. Our results demonstrate that modern network security requires a shift toward behavioral, sequence-aware intelligence. The high accuracy and low false-positive rates achieved by CyDetectNet prove its potential for real-world deployment in Security Operations Centers (SOCs). Future work will involve enhancing the model for real-time edge deployment and exploring Federated Learning to enable collaborative threat intelligence without compromising data privacy.

Future enhancements will include the integration of Attention Mechanisms to further improve the model's focus on critical packet fields and exploring Generative Adversarial Networks (GANs) for synthesizing new attack data to improve robustness against zero-day threats.

REFERENCES

[1] Kim, J., et al., 'Long short-term memory recurrent neural network for intrusion detection,' IEEE ICRAI, 2016.

[2] Kwon, D., et al., 'A survey of deep learning-based network anomaly detection,' Cluster Computing, 2019.

[3] Covington, P., et al., 'Deep Neural Networks for YouTube Recommendations,' Rec Sys, 2016.

[4] Fan, W., et al., 'Graph Neural Networks for Social Recommendation,' WWW, 2019.

[5] Zarzour, H., et al., 'Reconning: a recommender system using deep neural network with user and item embeddings,' ICICS, 2019.

[6] Sommer, R., and Paxson, V., 'Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,' IEEE S&P, 2010.

[7] Ying, R., et al., 'Graph Convolutional Neural Networks for Web-Scale Recommender Systems,' SIGKDD, 2018.

[8] Huang, Z., et al., 'TRec: an efficient recommendation system for hunting passengers with deep neural networks,' NCA, 2019.

[9] Hui, B., et al., 'Personalized recommendation system based on knowledge embedding and historical behavior,' Applied Intelligence, 2021.

[10] Hekmat far, T., et al., 'Embedding ranking-oriented recommender system graphs,' Expert Systems with Applications, 2021.

[11] Nareshkumar, R., and Nimala, K., 'An Exploration of Intelligent Deep Learning Models for Fine Grained Aspect- Based Opinion Mining,' ICSES, 2022.

[12] Sharafuddin, I., et al., 'Toward Generating a New Dataset for IDS,' ICISSP, 2018.

[13] Moustafa, N., and Slay, J., 'UNSW-NB15: a comprehensive data set for network intrusion detection systems,' MilCIS, 2015.

[14] Bengio, Y., et al., 'Representation Learning: A Review and New Perspectives,' IEEE TPAMI, 2013.

[15] Hochreiter, S., and Schmidhuber, J., 'Long Short-Term Memory,' Neural Computation, 1997.

[16] LeCun, Y., et al., 'Deep learning,' Nature, 2015.

[17] Goodfellow, I., et al., 'Deep Learning,' MIT Press, 2016.