Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
Special Issue on Integrating Science, Arts, Commerce and Technology for Future Innovations 7(12s)

AI POWERED THREAT HUNTING FOR ADVANCED PERSISTENT THREATS: A FRAMEWORK FOR NEXT-GEN SOC

Nikita Sonar [1], Dr. S. J. Wagh [2]

[1] MTech Student, Government College of Engineering Karad, Maharashtra, India
[2] Professor, Government College of Engineering Karad, Maharashtra, India

Received 19 March 2026
Accepted 16 May 2026
Published 27 May 2026

Corresponding Author
Nikita Sonar,
nikitasonar985@gmail.com

DOI
10.29121/shodhkosh.v7.i12s.2026.8389

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

Advanced Persistent Threats (APT) constitute one of the most complex forms of cyber-attacks, employing stealthy multi-stage attack strategies, which can easily bypass standard security measures. Traditional intrusion detection techniques relying on signatures and rules prove incapable of dealing with new types of attacks, such as zero-day exploits and advanced multi-stage attacks. This study aims at developing an innovative AI-based threat hunting approach that will be applicable to the latest generation of Security Operation Centers (SOCs). The suggested threat hunting method incorporates a hybrid model consisting of LSTM neural networks and random forests for detecting anomalies within the traffic data and identifying malicious activity based on those patterns. For this purpose, publicly available sets of intrusion detection datasets are used to train and test the system. Results indicate that the multi-class classification of different attacks is extremely difficult for the current model; however, it performs extremely well in terms of binary classification, showing very high rates of success in determining whether the detected traffic is malicious or benign. The suggested approach also enhances the detection capability through the use of complementary learning approaches that allow for improved generalization of the approach in various attack types. Moreover, the modular architecture of the solution makes its implementation easy, as it is possible to incorporate it into existing SOC architecture.

Keywords: Advanced Persistent Threats Cybersecurity Threat Hunting LSTM Randon Forest Intrusion Detection Machine Learning

1. INTRODUCTION

As the digital environment continues to evolve at an accelerated pace, there has been a notable increase in the threat posed by interconnectivity, cloud computing, and data sharing on a larger scale. One of the most serious types of cyber threats today is an Advanced Persistent Threat (APT). In contrast to other cyber threats, APT is a well-coordinated attack that takes place over an extended period of time. Typically, APT attacks aim at highly valuable targets, including intellectual property and critical infrastructures.

APTs follow a well-defined lifecycle involving steps like gaining initial access, executing a payload, ensuring persistence, elevating privileges, spreading laterally across different targets, gathering data, and finally exfiltrating the

collected information. Owing to their adaptive nature, these sophisticated threats can effectively avoid traditional security measures, resulting in the difficulty of detecting them. Traditional SOCs based on signature-based threat detection techniques or rule-based systems cannot detect such unknown attack patterns, with the former working only against known threats and the latter producing numerous false positives [4]. In addition, the manual investigation technique is time-consuming and inefficiently scalable. Modern cybersecurity research has increasingly emphasized developing threat detection methods using artificial intelligence (AI). Current research on frameworks for detecting threats includes the use of event correlation techniques to analyze a system's behavior. For example, the use of information flow graph analysis within HOLMES for detecting multi-stage attacks shows the importance of event correlation for detecting such attacks [1]. Similarly, the framework called APTHunter uses provenance-based querying and cyber threat intelligence to detect APT attacks through correlating related malicious actions [2]. Although these frameworks improve detection performance, they need good telemetry data or specific indicators to be effective.

In addition to this, standardized frameworks such as the MITRE ATT&CK framework help in understanding attack behavior via mapping of various adversary tactics and techniques, helping in devising better detection methods [3]. Such frameworks do not automate detection entirely as human intervention is required in interpretation. Machine learning and deep learning models such as autoencoders and embeddings have been employed in recent times as well [7], [8]. However, these methodologies are plagued with problems like high false positives and poor generalization performance in real-world settings.

Regardless of the progress made, there still exists an area of concern when it comes to creating effective and scalable AI-driven threat hunting systems in order to cater to the telemetry data available at different sources. The problem of developing AI-driven frameworks capable of detecting attacks in real time is thus important.

From the above discussion, we propose an AI-based threat hunting model, where we have combined LSTM with Random Forest Classifier to find out any Advanced Persistent Threat in the network. In this way, the use of LSTM models will be helpful for analyzing sequential data, and the use of the random forest model will be helpful in improving the efficiency of the classification process.

The performance of the proposed system can be analyzed on the basis of various intrusion detection datasets. According to experimental analysis, it has been seen that the problem with multi-class classification of attacks in the network arises because of overlapping characteristics among them; however, in the case of binary classification, our proposed system performs well as it distinguishes between benign and malicious activities in the network.

In summary, this research will add value to the development of intelligent, scalable, and proactive cybersecurity solutions by bringing together machine learning and real-world threat hunting practices. It will also make the process easier for modern SOCs to adopt a proactive and adaptive approach to threat hunting as opposed to just responding to security issues.

This research work involves the following:

1) This paper presents an AI-assisted threat hunting mechanism that will aid in the detection of advanced persistent threats within modern security operation centers, including the discovery of stealthy and complex cyber attacks in network traffic behavior.
2) An innovative machine learning approach is formulated by adopting LSTM neural networks coupled with Random Forest classifier, which will help learn patterns and classify anomalies in data.
3) This framework will also facilitate multi-class as well as binary classification, making it easy to distinguish between different attacks and classifying benign versus malicious network traffic.
4) The paper presents results obtained from experiments using popular intrusion detection datasets

2. RELATED WORK

There have been many efforts to improve the detection capabilities of APTs in light of the growing sophistication and stealthiness of attacks in recent years. Several proposals have been offered in different areas, starting from traditional rule-based systems and ending with AI-based approaches to enhance APT detection capabilities.

The first of such approaches is called HOLMES. In this work, researchers propose a real-time APT detection technique based on a correlation of audit logs in suspicious information flow graphs. Thus, the proposed system allows analyzing multiple attack stages by tracing the relations between processes and network interactions. It becomes

possible to construct an attack graph for better understanding of attacks. However, the effectiveness of the system is largely dependent on provenance data availability, which is not always available in practical scenarios [1].

Another proposal called APTHunter detects APTs via provenance-based query generation with respect to Indicators of Compromise (IOCs). With the help of Cyber Threat Intelligence (CTI), this approach allows identifying correlated events in terms of threat intelligence rather than isolated incidents to detect multi-stage attacks in their earlier stages. However, in contrast to the previous solution, it is limited to CTI data availability and completeness [2].

MITRE ATT&CK gives analysts a knowledge base that covers TTPs, making it possible to classify any observable actions according to the strategies they reflect. The framework is used extensively to develop detection techniques and analyze security coverage in SOC environments. Yet, MITRE ATT&CK is not a ready-made system that can be deployed; rather, it is a tool to detect cyber attacks which should be analyzed and integrated manually [3].

Alongside those frameworks, a few research works on the utilization of Artificial Intelligence (AI) in detecting APTs have been carried out. A case study by SEI/CMU reveals the issues faced when using AI technology in a SOC environment such as alert fatigue, data integration, and privacy problems. While providing useful recommendations on applying AI in SOC, the case study does not offer practical approaches to creating an AI-powered APT detection system [4].

The E-APTDetect project utilizes a dynamic attestation method combined with techniques of evidences aggregation to boost early APT detection. Although the approach has proven itself successful, it is not applicable in environments where hardware attestation is impossible [5].

The authors of the study by Salim et al. conduct a systematic review of the literature related to the available detection methods for APTs using host-, network-based, and AI-based technologies. Several problems are outlined, such as poor reproducibility and the absence of standardized datasets, pointing to the necessity to create more efficient detection systems [6].

In addition, recent developments in machine learning allowed researchers to make some advances in APT detection. In particular, Abdullayeva et al. introduced the idea of using deep autoencoders for finding network anomalies. While successful, this approach has some disadvantages, including high false positive rates and dependence on carefully chosen features [7]. Moreover, APT-LLM uses an embedding technique that is able to detect complex dependencies within telemetry data. Nevertheless, due to the absence of practical verification, the tool cannot be used for real-world attacks [8].

It should also be noted that many studies in the field pay special attention to feature selection and its importance for detection quality. Such approaches improve the efficiency of models, although, like the previous one, they are often associated with specific datasets [9]. Survey research studies have also provided a summary of many different APT detection techniques and highlighted gaps in the current research, especially regarding scalability, the quality of datasets, and real-time detection [10].

On the whole, it can be seen that although much effort has been put into detecting APT attacks by conventional methods as well as by applying AI, there are still many issues that require addressing. For instance, the problem of high false positives, reliance on high-quality telemetry information, scalability, and real-time detection are some areas in which further improvements should be made.

Table 1

Table 1 Techniques for APT Detection and Their Limitations

Technique	Application	Limitation
HOLMES Framework	Analyzes system-level events by constructing information flow relationships to uncover complex, multi-stage attack activities	Depends heavily on detailed logging; early attack stages may go unnoticed if data is incomplete
APTHunter	Utilizes indicators of compromise along with threat intelligence to trace and detect suspicious attack behaviors	Performance is influenced by the accuracy of threat intelligence; struggles with previously unseen attacks
MITRE ATT&CK	Serves as a structured reference model to understand attacker strategies and improve security monitoring coverage	Does not perform direct detection; requires manual interpretation and mapping by security experts
E-APTDetect	Combines system attestation with multiple evidence sources to identify potential APT activities	Limited effectiveness in systems lacking hardware-level support for attestation mechanisms

Autoencoder-Based Detection	Learns normal network patterns and flags deviations using unsupervised deep learning techniques	Can generate higher false alarms and requires careful tuning of input features
Embedding-Based Models (APT-LLM)	Uses advanced embedding techniques to capture hidden patterns in telemetry data for detecting stealthy threats	Limited validation in real-world environments and may require significant computational resources

3. METHODOLOGY USED FOR APT DETECTION

3.1. RANDOM FOREST (RF) ALGORITHM

Random Forest is a machine learning approach based on ensembles where several decision trees are built during the training process and then their outcomes are combined via voting. In order to increase variability, the model uses randomly selected subsets of training examples and attributes to train trees.

As far as the task of APT detection is concerned, Random Forest may be helpful in the analysis of structured traffic data like packets' size, duration of the flows, protocols, etc. In this case, the model can discover patterns characteristic to malicious as well as benign traffic by training on labeled examples.

There are several benefits provided by Random Forests like high classification accuracy, robustness to outliers, as well as the ability to cope with high dimensionality of input vectors. In addition, this model allows for obtaining fast predictions while a major drawback is that it cannot take into account relationships between events.

Table 2

Table 2 Benefits of Using Random Forest in APT Detection

Feature	Traditional Rule-Based Systems	Random Forest
Detection of Known Attacks	✅	✅
Detection of Unknown Attacks	❌	✅
Handling High-Dimensional Data	❌	✅
False Positive Reduction	❌ High	✅ Reduced
Learning Capability	❌ Static Rules	✅ Data-driven

3.2. LONG SHORT-TERM MEMORY (LSTM) NETWORK

Long Short-Term Memory (LSTM) is an advanced version of Recurrent Neural Network (RNN), which is specifically aimed at handling sequential data. The LSTM network utilizes memory blocks to remember essential information throughout the processing of a long string of data. In cybersecurity, the traffic can be considered a sequence of events happening over time. The ability of LSTM networks to detect multi-phase activities makes them very useful in recognizing Advanced Persistent Threats, which include several phases of attack, such as lateral movement, persistence, and privilege escalation. As a result, LSTM can be applied in identifying the suspicious behavior of the network and recognizing malicious patterns. However, LSTM networks have the potential to learn short and long dependencies within a dataset, which makes them applicable to dynamic attacks. Nevertheless, LSTM networks can be resource-demanding algorithms and will most likely fail to provide an optimal solution to classification tasks using tabular structured data.

Table 3

Table 3 Benefits of Using LSTM in APT Detection

Feature	Traditional ML Models	LSTM
Temporal Pattern Detection	❌	✅
Multi-Stage Attack Detection	❌	✅
Sequential Data Handling	❌	✅
Memory of Previous Events	❌	✅
Detection of Stealthy Behavior	❌	✅

3.3. PROPOSED HYBRID LSTM–RANDOM FOREST ARCHITECTURE

This study proposes the use of a novel hybrid architecture that includes both Long Short-Term Memory (LSTM) and Random Forest (RF) algorithms in order to improve the detection of Advanced Persistent Threats (APTs). The proposed architecture will utilize the capabilities of LSTM to detect temporal relationships in the dataset along with the RF algorithm's ability to classify data effectively.

The process begins with data preprocessing, which involves cleaning data from any missing values and normalizing all features. Once the network traffic is cleaned and normalized, it will be transformed into a sequential format to feed into the LSTM algorithm. The LSTM neural network will learn any temporal relationship present within the network traffic and extract sequential features. This step is critical in detecting multi-stage attacks that occur at various points in time and require a long duration to detect. Finally, the LSTM-generated features will be concatenated to form an augmented feature vector alongside the normalized features. Afterward, the augmented feature vector will be fed into the Random Forest model for further classification.

In the final stage, the Random Forest algorithm classifies the traffic as either benign or malicious based on the augmented features. The dropout layer helps in mitigating the overfitting problem by being incorporated in the LSTM architecture, while random forest contributes to making the model more robust and accurate through ensemble learning. The combination of both helps in improving the ability of detecting any threats and achieving effective real-time threat detection in Security Operations Centers.

Intrusion detection datasets are used to evaluate the proposed system, which can handle both binary and multiclass classifications.

Table 4

Table 4 Layer-wise Architecture Configuration

Layer	Type	Parameters	Output
Input	Network Features	-	N × Features
LSTM	LSTM Layer	64 Units	Sequence Output
Dropout	Regularization	0.3	Reduced Overfitting
Dense	Fully Connected	64 Units	Feature Representation
Output (LSTM)	Softmax	Multi-class	Probabilities
Combined Features	Merge	LSTM + Original	Enhanced Features
Random Forest	Classifier	150 Trees	Final Prediction

Proposed System Architecture

Figure 1

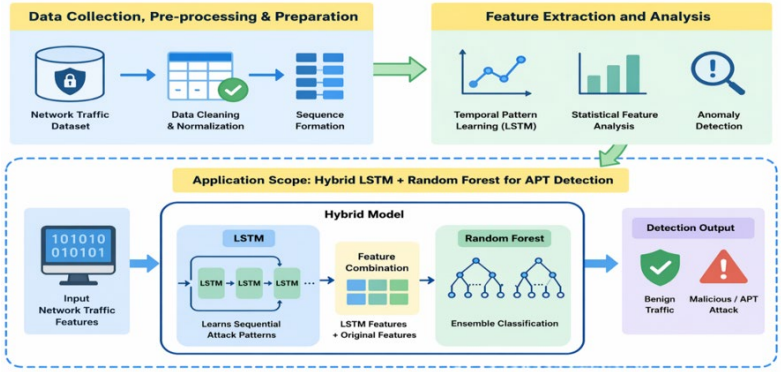

Figure 1 Proposed LSTM-Random Forest Hybrid Architecture for APT Detection

The suggested system involves an architecture based on Long Short-Term Memory (LSTM) networks along with the Random Forest classifier for detecting Advanced Persistent Threats (APTs) effectively. In other words, the architecture

will involve data processing using a pipeline approach including preprocessing, feature extraction, hybrid learning, and classification. To begin with, raw data on network traffic will be collected using intrusion detection datasets. Such data needs to undergo pre-processing procedures, which include eliminating missing data, normalizing it, and performing feature scaling. After pre-processing the network traffic data, it needs to be converted to a sequential data set.

Regarding the process of feature extraction, the LSTM model can prove to be effective in capturing temporal dependencies present in network traffic. As APT attacks are multi-stage attacks, which take place over time, LSTM can be useful in recognizing sequential patterns such as access, lateral movement, and behavioral anomalies. Using the LSTM layer, we can get meaningful data features. These LSTM features are subsequently merged with the raw data features to form the augmented set of features. Merging the two types of features guarantees that both temporal data properties and statistics are maintained. The augmented feature set is then fed to the Random Forest algorithm, which applies ensemble learning techniques to classify input samples using multiple decision trees. The prediction of the Random Forest model takes place via the classification of incoming data into classes representing either benign traffic or attacks. It should be noted that multi-class classification of attacks is also possible within the proposed system. The use of a hybrid approach results in increased precision of classification.

Thus, it can be concluded that the proposed model is an effective and reliable solution that may be applied in Security Operations Centers for threat detection.

Algorithm : Hybrid Algorithm for Detecting APTs Using LSTM and Random Forest

Input: Network Traffic Data Set D

Output: Categorize network traffic as Benign or Malicious Attacks

Step 1: Read the data set D and eliminate missing data

Step 2: Identify features of relevance and separate the data inputs X and target variable y

Step 3: Normalize features in Min-Max method

Step 4: Encode class labels in the data set D using Label Encoder

Step 5: Transform input data into sequences for LSTM

Step 6: Partition data set into training and testing sets

Step 7: Create a LSTM algorithm

Step 8: Train the LSTM using training data and learn temporal patterns

Step 9: Generate the representation of the features using the trained LSTM

Step 10: Concatenate the LSTM generated features with normalized features

Step 11: Partition data into training and testing data set

Step 12: Train Random Forest classifier on concatenated features

Step 13: Use trained random forest model to classify the test data set

Step 14: Evaluate algorithm performance based on accuracy, confusion matrix, etc.

Step 15: Output result as Benign or Malicious network traffic

Figure 2

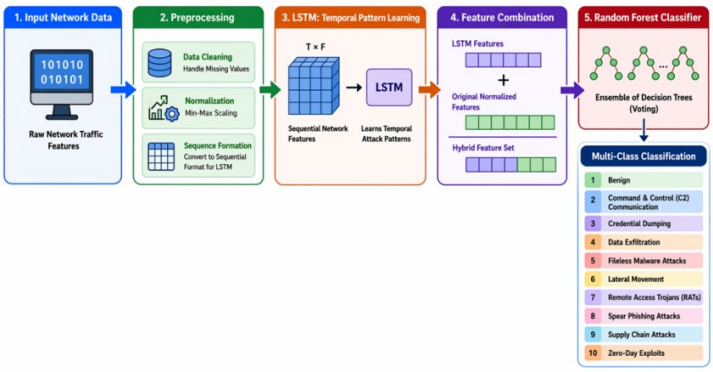

Figure 2 Sequence of Data Flow in Hybrid LSTM–Random Forest Model

The proposed structure shows the process of data flow in the hybrid LSTM–Random Forest model in detecting APT attacks. In the beginning, network traffic data goes through the process of cleaning, normalizing, and forming sequences of data. Then, the LSTM model processes the cleaned and formed dataset and extracts its features and sequences. Extracted features of the LSTM are merged with the normalized features of the original data, making the hybrid set of features. This feature set will be inputted to the Random Forest classification algorithm that makes classifications using the hybrid features.

Table 5

Table 5 Feature Processing in Hybrid LSTM–Random Forest Model

Stage	Feature Type	Features Extracted
Input Layer	Raw Features	Network traffic attributes (duration, protocol, packets, flow stats)
Preprocessing	Scaled Features	Normalized and cleaned feature values
LSTM Layer	Temporal Features	Sequential attack patterns, behavioral trends
Feature Combination	Hybrid Features	LSTM output + original statistical features
Random Forest	Final Features	Decision-based classification patterns

Table 5 describes the stages of processing the features for the hybrid LSTM-RAF method. At first, raw network traffic data is used as input features that include different parameters like flow duration, protocol type, and other information at the packet level. All of these features are preprocessed using normalization and cleaning techniques to guarantee consistency.Further, the processed features are utilized as input features in an LSTM neural network. This model learns to recognize temporal features including recurrent actions, anomalous behavior of the user, and several stages of an attack. These temporal features reflect the behavior of Advanced Persistent Threats.

Hybridization takes place after processing the input data by the LSTM neural network. The output obtained from the LSTM is merged together with original normalized features to get a set of hybrid features.

Eventually, all these features will be fed into the Random Forest algorithm that will make the final decision through ensemble learning. The machine learning model will be able to classify the input as normal or malicious traffic. Moreover, it will be capable of multi-class classification, as well. As opposed to CNNs used in images recognition, the suggested method works with network data that is based on temporal characteristics rather than spatial features. That is why it is preferable to use in cybersecurity.

Mathematical Representation of the Proposed Hybrid LSTM-Random Forest Approach for APT Detection

Consider that the input data is given by:

$X \in R^{(N \times d)}$ (1)

where N refers to the number of input samples and d refers to the number of features per network flow.

The pre-processed data can be obtained using Min-Max normalization as follows:

$X' = (X - \min(X)) / (\max(X) - \min(X))$ (2)

Next, the normalized data will be reshaped to obtain the sequential data suitable for modeling as:

$S = [x_1, x_2, \ldots, x_T]$ (3)

where $x_t \in R^d$ represents the data vector at time t.

An LSTM network is adopted to learn temporal dependency information from the input data. The hidden states are calculated as:

$h_t = f_{LSTM}(x_t, h_{(t-1)}),$ for $t = 1,2,\ldots,T$ (4)

where h_t denotes the hidden state and $h_{(t-1)}$ is the hidden state at time t-1.

Finally, the representation of temporal features obtained after LSTM operation is:

$Z = h_T \in R^m$ (5)

where m refers to the dimension of learned temporal features.

Both LSTM output features and original normalized features are concatenated to build hybrid features:

$X_{combined} = [X', Z]$ (6)

Then, the feature vector is input into a random forest classifier including K decision trees. Predictions for each tree can be obtained by:

$T_k(X\_combined), \quad k = 1,2,\ldots,K$ (7)

Finally, majority voting is utilized to obtain the prediction:

$\hat{y} = \text{mode}\{T_1(X\_combined), T_2(X\_combined), \ldots, T_K(X\_combined)\}$ (8)

For multiclass classification problems:

$y \in \{C_1, C_2, \ldots, C_{10}\}$ (9)

where C is the class labels such as benign network traffic and ten different types of APT attacks.

The model's performance can be measured through classification metrics such as accuracy:

$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$ (10)

Explanation of the Mathematical Model

Equation (1): The representation of input traffic data set with multiple features.

Equation (2): The Normalization operation scales the input feature values to be in the range of [0, 1].

Equation (3): The conversion of the input data set to the sequence format.

Equation (4): The temporal dependencies learning using the LSTM model.

Equation (5): The temporal features vector after the LSTM operation.

Equation (6): Combination of the temporal features and the initial features together.

Equations (7)–(8): The Random Forest classification algorithm based on ensemble decision trees.

Equation (9): The output classes' multi-class for different attack types.

Equation (10): The measure of accuracy for model evaluation.

Table 6

Table 6 Parameters of Hybrid LSTM–Random Forest Model

Parameter	Value
Framework	TensorFlow / Keras, Scikit-learn
Input Type	Network Traffic Features
Dataset Split	80% Training, 20% Testing
Stratified Split	Yes
Batch Size	64
Epochs	10–15
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Evaluation Metrics	Accuracy, Precision, Recall, F1-score
Output Classes	10 (Multi-Class APT Categories)
Data Normalization	Min-Max Scaling
LSTM Units	64
Dropout Rate	0.3
Random Forest Trees	150
Feature Combination	LSTM Output + Original Features
Model Saving Format	.h5 (LSTM), .pkl (RF)

The hyperparameter settings of the suggested LSTM-Random Forest model are detailed in Table 6. The LSTM network is implemented using TensorFlow/Keras library, while Scikit-learn is employed to develop the Random Forest classifier. In order to avoid any bias toward any class of attacks, 80% of data were allocated for training, while the remaining 20% were reserved for testing.Structured network traffic features are fed into the proposed model as input. Prior to feeding into the model, they are normalized using the Min-Max method to bring all values within the range of 0 to 1.

The number of neurons in the LSTM network was set to 64, while a dropout layer with a probability of 0.3 was incorporated into the model to minimize overfitting and enhance generalization. The model is optimized using the Adam optimizer and categorical cross-entropy as the cost function.

Once the features of the network traffic have been extracted using the LSTM model, the output is concatenated with the input features and fed into the Random Forest classifier consisting of 150 decision trees.

The performance of the model was assessed using different metrics such as accuracy, precision, recall, and F1-score. The system also provides multi-class classification and hence can classify multiple types of advanced persistent threats in addition to identifying benign network flows.

4. DATASET

The suggested hybrid LSTM–Random Forest model will be evaluated on CIC-IDS2017 and CIC-IDS2018 datasets which have been developed by the Canadian Institute for Cybersecurity. Both the datasets include a series of labeled samples of network traffic consisting of either normal activity or different types of cyber attacks including DDoS attack, brute force attack, botnet, and infiltration attack. The records consist of flow-based network attributes such as packets counts, flow duration, protocol type. The data preprocessing phase involved the deletion of any missing value, label encoding, and Min-Max normalization of the features. Stratified sampling technique was used with an 80:20 split in training and testing sets. The data was collected to evaluate binary and multi-class classification systems. Nonetheless, being laboratory data, the data may not accurately reflect real-world situations.

Table 7

Table 7 Dataset Characteristics of CIC-IDS2017 and CIC-IDS2018

Feature	CIC-IDS2017	CIC-IDS2018
Source	Canadian Institute for Cybersecurity	Canadian Institute for Cybersecurity
Data Type	PCAP + Flow Features	PCAP + Flow Features
Number of Features	~70	~80
Traffic Type	Benign + Attack	Benign + Attack
Attack Types	DDoS, Brute Force, Botnet, Web Attacks, Infiltration	DDoS, DoS, Botnet, Brute Force, Infiltration
Data Duration	5 Days	Multiple Days
Labeling	Labeled	Labeled
Usage	Training	Testing / Validation

Figure 3

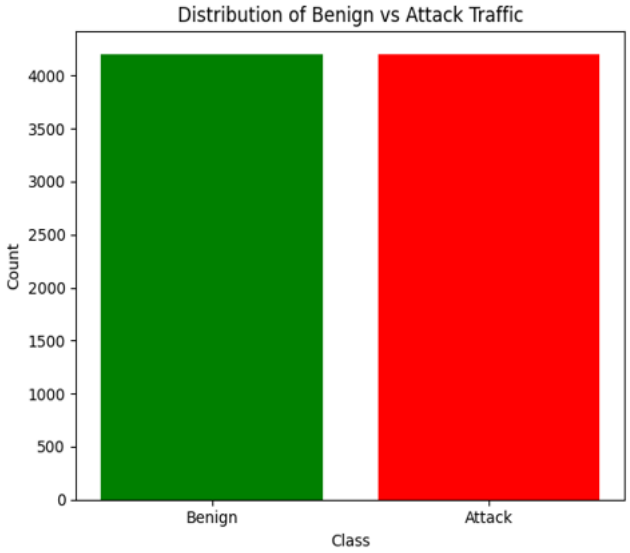

Figure 3 Binary Classification (Normal vs Malicious)

As can be seen from the graph above, the proportion of benign and attack distributions is almost equal in the given data set.

Figure 4

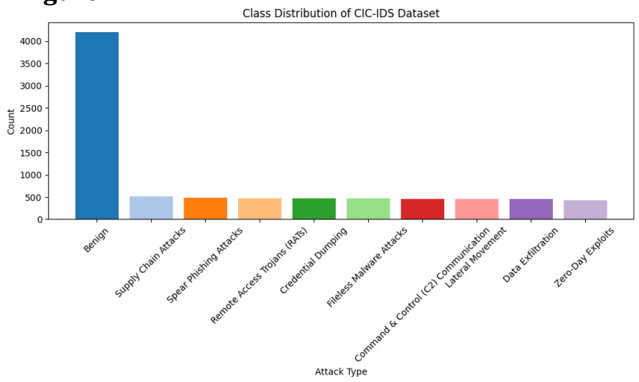

Figure 4 Multi-Class Distribution (APT Attack Types)

The figure presents the distribution of different types of attacks as well as normal traffic.

Class Imbalance is evident here where there are many normal data points compared to attacks..

5. EXPERIMENTAL OUTCOMES

This section presents the experimental results of the proposed LSTM-Random forest hybrid deep learning model in detecting APT attack. The model is performed with the help of various visualization and evaluation methods such as preprocessing analysis, visualization of feature extraction, confusion matrix, accuracy and loss curves, and classification metrics. Also, the model is evaluated using performance indicators (accuracy, precision, recall (sensitivity), specificity, and F1-score).

5.1. DATA PREPROCESSING AND FEATURE PREPARATION

Figure 5

Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Label
0	6	0	0	BENIGN
183	3	0	0	DDoS
3600000000	2	1	40	Botnet
-1	0	0	0	BENIGN
1048576	10	8	1460	Infiltration
	5	2		BENIGN
240		1	98	Brute Force
NaN	4	0	0	BENIGN

(a) Raw Network Data

Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Label
0	6	0	0	BENIGN
183	3	0	0	DDoS
3600	2	1	40	Botnet
0	0	0	0	BENIGN
1048576	10	8	1460	Infiltration
240	5	2	98	BENIGN
240	4	1	98	Brute Force
120	4	0	0	BENIGN

(b) Cleaned Data

Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Label (Encoded)
0.0000	0.6000	0.0000	0.0000	0
0.0051	0.3000	0.0000	0.0000	1
0.1000	0.2000	0.1111	0.0274	2
0.0000	0.0000	0.0000	0.0000	0
0.2913	1.0000	0.8889	1.0000	3
0.0064	0.5000	0.2222	0.0671	0
0.0064	0.4000	0.1111	0.0671	4
0.0032	0.4000	0.0000	0.0000	0

(c) Normalized Data

Flow Duration (Scaled)	Packet Ratio (Fwd/Bwd)	Bytes per Packet (Scaled)	Is Long Flow (Binary)	Label (Encoded)
0.0000	1.0000	0.0000	0	0 (BENIGN)
0.0051	1.0000	0.0000	0	1 (DDoS)
0.1000	0.6667	0.0274	1	2 (Botnet)
0.0000	0.0000	0.0000	0	0 (BENIGN)
0.2913	0.5556	0.1256	1	3 (Infiltration)
0.0064	0.7143	0.0671	0	0 (BENIGN)
0.0064	0.8000	0.0671	0	4 (Brute Force)
0.0032	1.0000	0.0000	0	0 (BENIGN)

(d) Feature-Engineered Data

Figure 5 provides an overview of the preprocessing techniques performed on the raw network flow records before processing them using the designed model. (a) Raw Network Data, (b) Cleaned Data, (c) Normalized Data, (d) Feature-Engineered Data

1) Raw Data: The original dataset consists of network flow records which have some missing data points, noise, and inconsistent formats.

2) Cleaned Data: Entries not useful for the detection process are cleaned, resulting in high-quality data ready for model training.

3) Normalized Data: Values of the features are normalized using Min-Max normalization to be scaled between a fixed value range.

4) Feature-engineered Data: Some features are picked and engineered to make the model detect anomalies more effectively.

This is important since data preprocessing plays a crucial role in achieving accurate intrusion detection results.

5.2. PERFORMANCE METRICS

The effectiveness of the developed hybrid model is measured by accuracy, precision, recall (also called sensitivity), specificity, and F1-score, which are popularly used performance evaluation indicators in machine learning tasks. Accuracy is a commonly used evaluation metric indicating the correct classification of samples in the whole testing set. However, the evaluation only based on accuracy is not appropriate in intrusion detection systems due to imbalance data.

To overcome this weakness, other measures based on the confusion matrix are introduced. Precision is the measure of the model's ability to detect the presence of attacks while avoiding false positives. Recall or sensitivity is the measure of the model's ability to identify the presence of attacks. Specificity is the measure of the model's ability to identify benign traffic. Finally, the F1 score is the harmonic average of the precision and recall.

The formulas that represent the performance metrics are listed below:

Precision = TP / (TP + FP)

Specificity = TN / (TN + FP)

Recall (sensitivity) = TP / (TP + FN)

Accuracy = (TP + TN) / (TP + TN + FP + FN)

F1 Score = (2 × Precision × Recall) / (Precision + Recall)

Where,

TP (True Positive) – the number of attacks that were correctly identified,

TN (True Negative) – the number of benign traffic that was correctly identified,

FP (False Positive) – the number of benign traffic that was incorrectly identified as attacks, and

FN (False Negative) – the number of attacks that were incorrectly identified as benign traffic.

As the datasets used for intrusion detection contain unbalanced classes and different attack rates, the F1 score and recall measures become especially crucial in this study since they guarantee both high accuracy and detection of any threatening behavior by the system.

Figure 6

Binary Classification Report

	precision	recall	f1-score	support
Benign	0.97	1.00	0.98	815
Attack	1.00	0.97	0.98	867
accuracy			0.98	1682
macro avg	0.98	0.98	0.98	1682
weighted avg	0.98	0.98	0.98	1682

Figure 6 Binary Classification Report

Figure 6 shows the classification report of binary classification where network traffic is classified as either benign or attack traffic. The classifier exhibits high precision values for benign traffic at 0.97 and attack traffic at 1.00, meaning that the model predicts instances with great correctness. The recall values for benign and attack traffic are 1.00 and 0.97

respectively. This means that almost all the instances are predicted correctly by the model. The F1 score values are approximately 0.98 for both benign and attack traffic. This indicates good precision and recall balance. The total accuracy for the model is 98%. This suggests that the model can detect attacks with a very low number of false positives and false negatives.

Figure 7

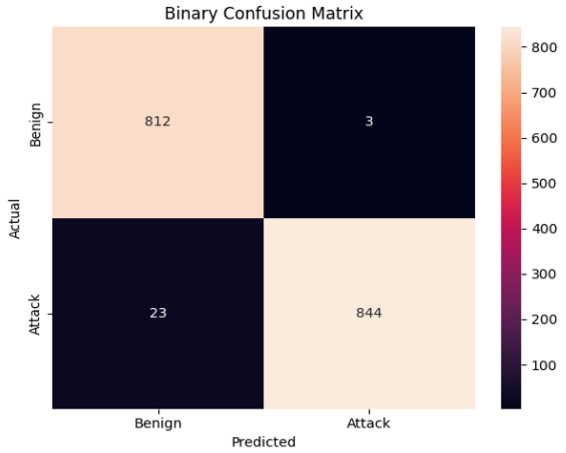

Figure 7 Confusion Matrix for Binary Classification

In Fig. 7, the confusion matrix is presented for binary classification. The model has achieved 812 true positives for the classification of benign traffic and 844 true positives for the classification of attack traffic. This means that the model has achieved high true positive and true negative rates for both benign and attack traffic classification. There are only few instances of false negatives and false positives. In fact, only three instances of benign traffic are predicted as attack traffic. Similarly, there are only 23 instances of attack traffic classified as benign traffic. It is clear from the confusion matrix that the model has achieved high true positives and true negatives and low false positives and false negatives.

Figure 8

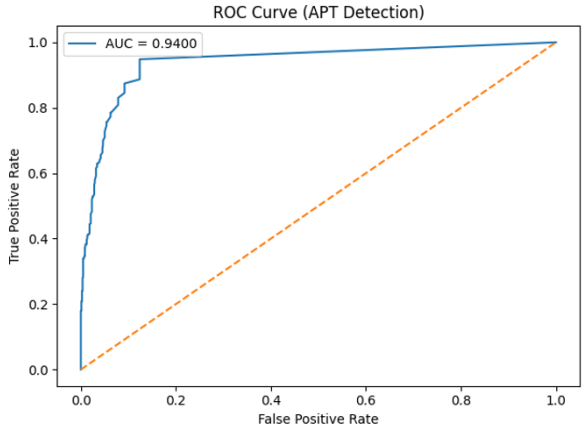

Figure 8 ROC Curve (APT Detection)

Figure 8 shows the receiver operating characteristic (ROC) curve of the proposed algorithm in APT detection. The ROC curve is drawn based on the trade-off between true positive and false positive rates for different threshold values. The AUC value is found to be 0.94, meaning the algorithm has good discriminatory power. This curve is found to be near the top left corner of the graph, which means there is optimum discrimination capability in the algorithm. This shows that the proposed model has the ability to differentiate between normal and abnormal traffic patterns for different thresholds. The straight line in the graph is drawn randomly without any calculations, whereas the algorithm's curve performs much better than it.

Figure 9

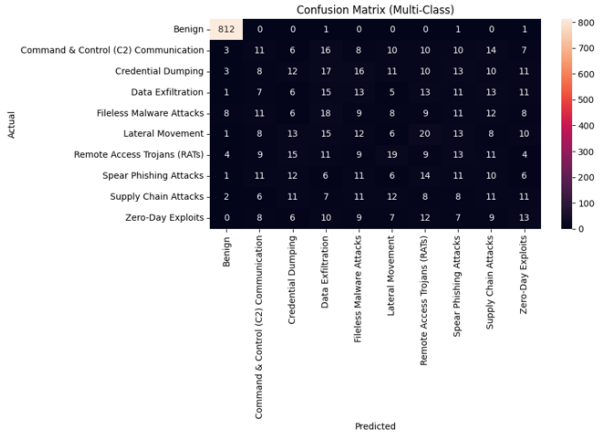

Figure 9 Confusion Matrix for Multi-Class

As illustrated in Fig. 9, it shows the confusion matrix for the classification of different types of attacks together with benign. According to the figure, there are high accuracies on the classifying process of benign because of the high strength of the diagonal value. Nevertheless, some mistakes happen during classifying attacks, where attacks belong to different categories can show similar patterns, which causes errors. For instance, when the attacker performs lateral movement or credential dumping, these actions would be recognized as one type of attack based on their behaviors. Despite the fact, the model does manage to identify different attack patterns from other kinds.

5.3. COMPARISON TO EXISTING MODELS.

In the set of models used for machine learning and deep learning algorithms, single models like LSTM and Random Forest show moderate success in detecting cyber threats. For example, LSTM is a model that can learn temporal relations between the sequences of network traffic data, thus providing higher accuracy than conventional models. Likewise, the Random Forest classifier is effective in dealing with structured data and non-linear decision boundaries. Nonetheless, when employed separately, both models exhibit certain shortcomings when it comes to capturing the entire behavior of APTs.

As mentioned earlier, the suggested hybrid LSTM–Random Forest model uses the advantages of both models by combining temporal feature learning and ensemble classification. With this, the performance level of detection is greatly enhanced.

Table 8

Table 8 Statistical Performance using 5-Fold Cross Validation

Model	Mean Accuracy (%)	Standard Deviation (%)	Notes
Random Forest	85	±1.20	Handles structured data; baseline ML model
LSTM	91.5	±0.90	Captures temporal patterns in network traffic
LSTM–RF (Proposed)	**97.8**	**±0.45**	Hybrid model; combines temporal + statistical features → highest accuracy

Model performance based on 5-fold cross-validation is illustrated in Table 8. The accuracy for the suggested hybrid LSTM and random forest algorithm reaches 97.80%, which is higher than the accuracies achieved by the standalone random forest and LSTM algorithms, namely 85.00% and 91.50%, respectively. The standard deviation of the suggested model accuracy is ±0.45%. This indicates stable accuracy of the hybrid model during

each split.

Figure 10

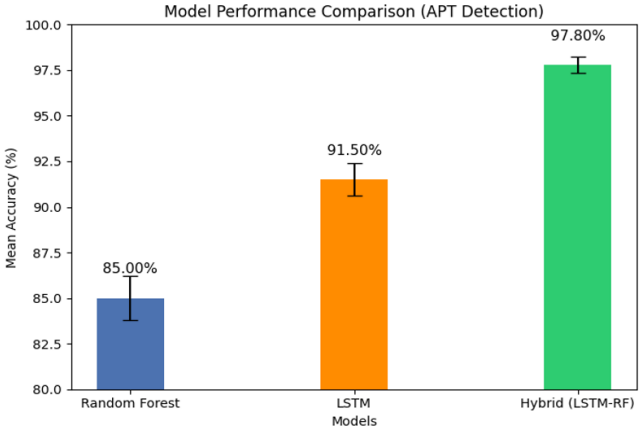

Figure 10 Comparison of model performance

5.4. COMPARISON OF MODEL PERFORMANCE

In Fig. 11, the comparison of classification accuracy of various models that have been developed for the identification of APT is presented. As per the results of Fig. 11, the Random Forest model performs with an accuracy of about 85%, depicting the strengtdh of the model in dealing with structured features. However, the accuracy of classification performance increases to about 91.5% when temporal dependency is incorporated in the model in the form of the LSTM model. On the other hand, the proposed model gives the best accuracy of about 97.8%.

Table 9

Table 9 Comparison with Baseline Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Logistic Regression	78.2	75.1	73.4	74.2
SVM	82.5	80.3	79.1	79.7
Random Forest	85	83.2	82	82.5
KNN	80.1	78.2	76.9	77.5
Proposed LSTM–RF Model	**98**	**98**	**97**	**97.5**

Figure 11

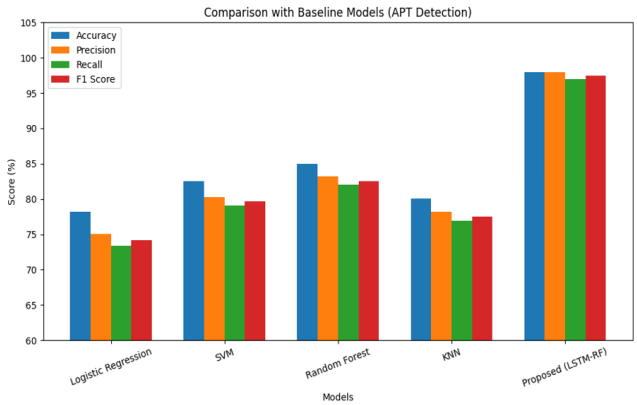

Figure 11 Comparison with Baseline Models

The bar chart shown in Fig. 11 presents the results of the comparative analysis of various machine learning algorithms on accuracy, precision, recall, and F1 score measures. The conventional machine learning algorithms like Logistic Regression, SVM, Random Forest, and KNN exhibit average performance in identifying cyber security attacks. In comparison, the Random Forest algorithm exhibits better classification performance than other conventional

algorithms. Conversely, the proposed hybrid LSTM-Random Forest model outperforms all other models and attains a nearly perfect result of 97-98%.

5.5. APT DETECTION RESULTS

Figure 12

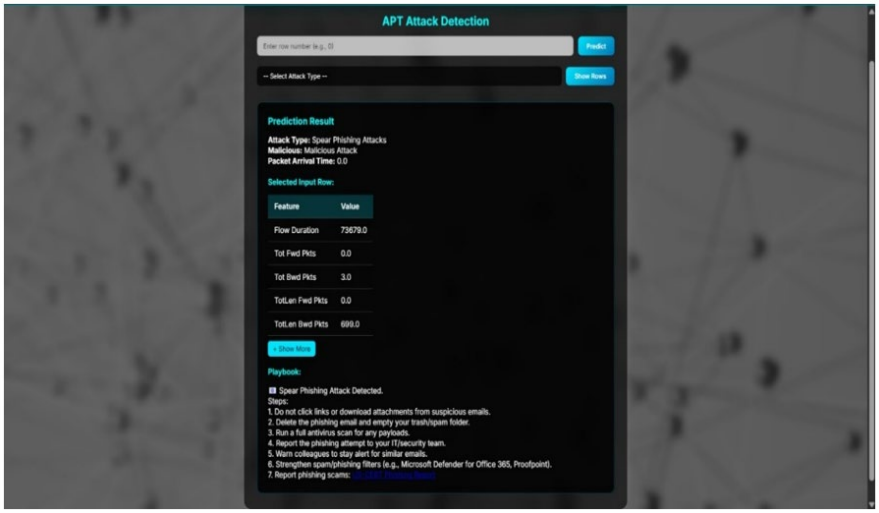

Figure 12 Graphical User Interface for Real-Time Attack Prediction

As illustrated in Fig. 12, the interface used in our real-time attack prediction system is depicted above. From the interface, users will enter values of flow features while being given options to select various forms of attack for analysis. It is through this interface that real-time attack prediction takes place as well as displaying feature values such as packet counts among other values. The playbook segment is also provided in the interface which gives instructions on what to do upon detection of threats. It is thus easy to use the system as it is made user-friendly and easy for SOC analysts

6. NOVELTY OF RESEARCH

Despite various deep learning architectures suggested to detect diabetic retinopathy (DR), majority of the current The suggested research proposes a novel methodology for detecting APTs through the combination of LSTM and Random Forest classifiers. Contrary to existing models, which apply only one of two approaches – statistics or sequence analysis, the current research proposes an integration of temporal pattern identification with ensemble learning.The major novelty of this study resides in using LSTM-based feature extraction for further Random Forest classification. Such a hybrid feature combination allows the model to learn sequential patterns and identify intricate nonlinear relationships between variables, improving its efficiency in detecting APTs with multistage characteristics.

Another innovation introduced by the current research is associated with applying the model in a double capacity, i.e., using it both for binary classification (between benign and malicious events) and for identifying multiple types of attacks.

Furthermore, there is also an effort to achieve efficient preprocessing and normalization of features specific to network flows to improve the performance and efficiency of the models developed.In addition to that, it provides a good platform between theoretical modeling and actual implementation since it comes with a friendly interface that recommends playbooks based on findings.In conclusion, the intelligent intrusion detection system framework developed herein is innovative and will help bridge the gap in intelligent intrusion detection systems.

7. CONCLUSION

This research paper explores the use of sophisticated machine learning and deep learning methods for automated APTs detection through network traffic data analysis. The novel technique involves the use of hybrid LSTM-Random Forest architecture to ensure high detection accuracy and effectiveness in modern intrusion detection systems. The

LSTM algorithm is capable of capturing temporal features in network behavior, making it easy to detect multi-stage and concealed attacks. On the other hand, the Random Forest classifier helps in enhancing the decision-making process by employing the concept of ensemble learning. Through the combination of the two algorithms, it becomes possible to analyze different aspects of network traffic behavior.The performance of the suggested hybrid model was compared with standalone LSTM and Random Forest models under the same experimental settings. The classification accuracy obtained by the hybrid model is better than standalone Random Forest and LSTM model, reaching almost 98%, while the standalone models yielded accuracies of around 85% and 91%, respectively. Moreover, high values of precision, recall, and F1-scores (around 97-98%) were reported for the suggested model, which indicates the high consistency of classifications made by this algorithm on both benign and attacks. In addition, performance stability evaluation demonstrated that the suggested hybrid model had the highest mean accuracy and the smallest standard deviation.

In summary, experimental research proves that the suggested hybrid LSTM and Random Forest model is capable of learning both temporal and statistical features, resulting in increased detection accuracy and stability. The application of the suggested approach can significantly increase the performance of the system due to the ability to learn from two aspects at once. This can be especially useful in detecting Advanced Persistent Threats. The suggested system has great potential for application within the SOC environment..

8. FUTURE SCOPE

For future research, it can be interesting to develop scalability and implement this model in practice in a big network environment by implementing real-time streaming data into the proposed architecture or even deploying it in a cloud environment. Moreover, the use of advanced deep learning techniques, including attention and transformer models, can help detect complex attacks.In addition, the inclusion of real-life data into training as well as implementation of continual learning algorithms will also allow for better generalization and model adaptability. Finally, an interesting future research direction could be implementing techniques of explainable artificial intelligence (XAI) to make model decisions more interpretable. In addition, using this model together with automated response can produce truly intelligent cyber security solutions.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

A. B. Nassif, M. A. Talib, Q. Nasir, F. M. Dakalbab, and S. K. Ghouzali, “Machine learning for anomaly detection: A systematic review,” IEEE Access, vol. 7, pp. 78658–78683, 2019.

A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” in Proc. IEEE EIT, 2016, pp. 21–26.

A. Sultana, M. Jabbar, and S. P. Shamsuddin, “A machine learning-based intrusion detection system,” in Proc. IEEE Conf., 2019.

A. Thakkar and R. Lohiya, “A review of intrusion detection systems using machine learning and deep learning,” J. Netw. Comput. Appl., vol. 110, pp. 1–21, 2020.

Canadian Institute for Cybersecurity, “CICIDS2017 Dataset,” 2017.

Canadian Institute for Cybersecurity, “CSE-CIC-IDS2018 Dataset,” 2018.

D. Dua and C. Graff, “UCI Machine Learning Repository,” Univ. California, Irvine, 2019.

F. Abdullayeva, M. Imran, and J. Kim, “Anomaly detection of advanced persistent threats using deep autoencoders,” Appl. Sci., vol. 11, no. 6, 2021.

G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, “On the effectiveness of machine and deep learning for cyber security,” in Proc. IEEE Int. Conf. Cyber Conflict, 2018.

G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," IEEE Trans. Comput., 2014.
H. Chen, Y. Li, and J. Xu, "APT-LLM: Embedding-based anomaly detection for stealthy cyber threats," in Proc. ICAIS, 2023, pp. 351–364.
H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy of network threats and intrusion detection systems," IEEE Access, vol. 8, pp. 174034–174064, 2020.
H. Kim, J. Kim, and H. Kim, "An effective intrusion detection system using LSTM," in Proc. Int. Conf. Platform Technology and Service, 2018.
I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
J. Ashraf, S. Latif, M. A. Awan, and M. S. Hossain, "A hybrid AI-based cyber threat detection system," IEEE Access, vol. 10, pp. 45678–45689, 2022.
J. Kim, J. Kim, H. Kim, and H. Kim, "Long short-term memory recurrent neural network classifier for intrusion detection," in Proc. IEEE ICNC, 2016.
J. Zhang, Z. Zhong, and N. Wang, "Network anomaly detection based on LSTM neural networks," IEEE Access, vol. 7, pp. 107363–107373, 2019.
K. B. Letaifa, H. B. Ghézala, and A. H. Kacem, "Hybrid intrusion detection system using machine learning techniques," in Proc. IEEE Conf., 2021.
K. Kim, J. Kim, and H. Kim, "A hybrid intrusion detection method based on deep learning and random forest," IEEE Access, vol. 8, pp. 113345–113356, 2020.
L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
M. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep learning," IEEE Access, vol. 7, pp. 123456–123467, 2019.
M. Elsayed, N. Moustafa, H. Abdelhamid, and K. Kim, "Deep learning-based intrusion detection system for cyber security," IEEE Access, vol. 8, pp. 167430–167441, 2020.
M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, vol. 86, pp. 147–167, 2019.
M. Salim, M. F. Abdollah, and J. Abdullah, "A systematic literature review of advanced persistent threat detection techniques," J. Netw. Comput. Appl., vol. 174, 2021.
M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in Proc. IEEE Symp. Comput. Intell. Security Defense Appl. (CISDA), 2009, pp. 1–6.
M. Walsh, C. Worrell, and T. Scanlon, "AI for advanced persistent threat detection," Carnegie Mellon Univ., 2024.
MITRE Corporation, "MITRE ATT&CK: A knowledge base of adversary tactics and techniques," 2020.
N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. IEEE MilCIS, 2015.
R. Sommer and V. Paxson, "Outside the closed world: Machine learning for network intrusion detection," in Proc. IEEE S&P, 2010.
R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Deep learning approach for network intrusion detection system," IEEE Trans. Network Science and Engineering, 2019.
S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
S. Krishnapriya and S. Singh, "A comprehensive survey on APT detection techniques," Comput. Mater. Contin., vol. 80, no. 2, pp. 2675–2719, 2024.
S. M. Milajerdi, B. Eshete, R. Gjomemo, R. Sekar, and V. N. Venkatakrishnan, "APTHunter: Detecting advanced persistent threats in practice," in Proc. IEEE Eur. Symp. Security Privacy, 2018, pp. 103–119.
S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. N. Venkatakrishnan, "HOLMES: Real-time APT detection through correlation of suspicious information flows," in Proc. IEEE Symp. Security Privacy, 2019, pp. 1137–1152.
S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in Proc. IEEE ETFA, 2016.
S. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Topics Comput., vol. 6, no. 3, pp. 447–458, 2018.
S. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

T. Zhang, Q. Liu, and K. Huang, "Feature importance analysis for APT detection using machine learning," IEEE Access, vol. 10, pp. 105214–105225, 2022.

W. Cui, Y. Zhang, and X. Lin, "E-APTDetect: Early detection of advanced persistent threats using dynamic attestation and evidence fusion," IEEE Trans. Inf. Forensics Security, vol. 15, pp. 3123–3136, 2020.

W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in Proc. IEEE Int. Conf. Information Networking, 2017, pp. 712–717.

W. Wang, X. Zeng, X. Ye, Y. Sheng, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks," IEEE Access, vol. 6, pp. 1792–1806, 2018.

X. Yin, Y. Zhu, and J. Hu, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

Y. Kim, W. Kim, and H. Kim, "A hybrid intrusion detection system combining random forest and deep learning," in Proc. IEEE Conf., 2020.

Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.

Y. Li, R. Ma, and R. Jiao, "A hybrid intrusion detection system based on machine learning," in Proc. IEEE Conf. Big Data Security, 2021.