Real-time Threat Analysis and Improving Cybersecurity Defenses in Evolving Environments with Deep Learning and Traditional Machine Learning Algorithms

Md. Anisur Rahman
Department of computer science and engineering, Jahangirnagar University, Dhaka, Bangladesh

Md. Sahidullah
Department of computer science and engineering, Asian University of Bangladesh, Dhaka, Bangladesh

Farjana Kamal Konok
Department of computer science and engineering, Asian University of Bangladesh, Dhaka, Bangladesh

ABSTRACT

Real-time threat analysis plays a critical role in modern cybersecurity, ensuring that systems remain protected against evolving cyber threats. This study aims to develop and evaluate a robust model for threat detection using a combination of deep learning and traditional machine learning algorithms. The proposed methodology employs deep learning techniques alongside traditional algorithms, leveraging a comprehensive threat detection dataset for training and validation. The model achieved the highest accuracy of 97% with minimal loss, converging efficiently within the initial training epochs. Results indicate that the model achieved reliable generalization with close alignment between training and validation performance, showcasing its effectiveness in detecting threats accurately. The contributions of this study lie in advancing cybersecurity mechanisms through the integration of machine learning models, paving the way for enhanced real-time threat detection and response. Future enhancements, including advanced architectures such as Transformers, are proposed to further improve performance and applicability across broader cybersecurity domains.

General Terms

Real-time Threat Analysis with machine learning integration.

Keywords

real-time threat analysis, cybersecurity, machine learning, deep learning, threat detection, model accuracy, Transformers, traditional algorithms, dataset performance

1. INTRODUCTION

The escalating sophistication and frequency of cyber-attacks present new threats to our interconnected globe and digital environments. As the digital built environment expands, so do the vulnerabilities, resulting in severe repercussions for individuals and organizations [1]. Due to the changing landscape where conventional methods are inadequate, cybersecurity experts are tasked with identifying, mitigating, and preventing real-time threats. Advanced threat intelligence, grounded in timely computational techniques, is now essential for safeguarding sensitive information and ensuring system functionality [2], [3]. The contemporary threat landscape is marked by dynamic, developing cyber threats that necessitate more proactive defense techniques than classic static measures can offer.

Conventional cybersecurity systems frequently employ signature-based detection, which lacks the agility to react to the dynamic nature of threats. Nevertheless, current methodologies fail to generalize to novel threats, hence exposing systems to the danger of zero-day attacks and other sophisticated exploitation techniques [4]. Additionally, the synergistic potential of digital ecosystems can create new issues by bringing heterogeneous systems together, exposing exposure to systemic risk [5]. Thus, we now require real-time, adaptive solutions more than ever. Emergent AI-driven methodologies, encompassing sophisticated deep learning (DL) and traditional machine learning (ML) algorithms, have demonstrated significant potential to surmount these obstacles [6], [7]. Nonetheless, recent research has generally focused on the standalone utilization of such technologies, whereas the synergistic deployment of these technologies within an integrated model for real-time threat analysis has received less attention [8], [9].

This research aims to bridge the gap by developing a hybrid framework that combines the strengths of deep learning and traditional machine learning algorithms. By leveraging their complementary capabilities, the proposed approach seeks to enhance real-time threat detection in dynamic and evolving environments. Unlike existing methods, this hybrid framework is designed to address both known and unknown threats effectively, thereby improving the overall cybersecurity defense mechanism [10], [11]. The framework's real-time capabilities are essential for detecting anomalies, mitigating attacks, and maintaining the resilience of digital ecosystems.

We have produced four important contributions through our research. First, we offer a new hybrid architecture that blends deep and standard machine learning algorithms in order to better real-time attack detection. Secondly, it gives a realistic performance assessment of the suggested framework, thereby rating its utility and robustness. Third, it emphasizes the merits and downsides of both deep learning and classical machine learning methods by comparing these approaches. Lastly, it sets the basis for future improvements in AI-driven cybersecurity systems, bringing insights into fields including adversarial attacks, ransomware detection, and collaborative threat intelligence [12], [13], [14], [15].

The rest of this paper is organized as follows: Related work is given in Section I, revealing gaps as well as limitations inherent in existing research. Section II covers the methodology section, where we discuss the proposed hybrid framework and algorithms. Section III reports on the results and discussion of the performance evaluation with different models. Section 4 elaborates on the ramifications of the discoveries and provides prospective avenues for future study. Section V finally finishes the study, outlining the essence of its contributions and possible applications.

2. LITERATURE REVIEW

given the DT, we predict ML (and DL as well) methodologies and talents to be a spectacular revolution in cybersecurity—its application compared to the complexity, variety, and volume of cyber threats is a crucial area of research for data scientists and practitioners. Traditional machine learning methods based on designed features and supervised learning methods have been widely employed for intrusion detection systems (IDS) and threat detection issues. Akcay and Breckon [16] reviewed recent improvements in X-ray security imaging and marked the necessity of disclosing the progress of machine learning to better automate procedures to detect a danger, such as those found in a high-risk atmosphere (i.e., airports). In a similar research work, Liu and Lang [17] performed a review of the traditional ML and DL techniques for IDS, emphasizing that traditional ML methods (like decision trees, support vector machines (SVM), and k-nearest neighbor (KNN)) have shown good performance, but they are often limited by their reliance on manual feature extraction and cannot scale well to jobs associated with immense data. The pioneering work of Martínez Torres et al. Machine learning techniques have been demonstrated by [31] to be significant in numerous applications of cybersecurity; this study provides the machine learning techniques, along with several use cases, such as anomaly detection and malware categorization.

Deep Learning Hits the Scene: Deep learning is the latest kid on the block and helps us to transcend the constraints of classical machine learning by employing neural networks. Models based on deep learning approaches, for example, CNNs, RNNs, and autoencoders for high-dimensional data and the capture of complex patterns, produce good results. Su et al. [18] established that DL models outperformed classic ML techniques in terms of accuracy and scalability for the purpose of network intrusion detection, which was illustrated using the NSL-KDD dataset. Vinayakumar et al. They observed that RNNs and their derivatives have a wide range of applications in intrusion detection systems due to their capacity to deal with sequential data, which is valuable in the context of real-time detection of adaptive threats. An in-depth study of DL applications in the cybersecurity area has been published by Alazab and Tang [23], highlighting the potential of the described models in malware detection, phishing attack mitigation, and fraud detection. In another research study, Dixit and Silakari [24] emphasized technology and the status of DL algorithms utilized in cybersecurity, making major breakthroughs from neural networks for applications like spam filtering and botnet identification. Halbouni et al. [25] offered a summary of the deployed models, pointing out the problems confronted with DL actual situations like imbalance, overfitting, and computational expenditures.

To address the limits of individual techniques, hybrid approaches that blend classical ML, DL, and other computational methods have arisen as a possible answer. These approaches integrate the advantages of both methodologies to promote precision, extensibility, and responsiveness in changing circumstances. Smys et al. In [20], a hybrid intrusion detection system is provided, focusing solely on the threshold of IoT contexts, which also tackles the needs of resource-limited devices and diverse forms of attacks. Ieracitano et al. In the area of intelligent intrusion detection, [21] proposed a framework that integrates statistical analysis with autoencoders and produced exceptional results in terms of detection rates and false positives. The work by Ferrag et al. Application of FDL [29] indicated that the use of decentralized learning methodologies can lead to greater privacy preservation and data-sharing security in IoT networks. Similarly, Itasoy et al. The system in [37] is a hybrid that combines isolation forest with XGBoost for the goal of detecting suitability for specific security encounters. Chirra's method [33], particularly, blends traditional and DL models to improve cybersecurity, specifically in hybrid cloud environments, and further extends this concept of AI-based real-time security monitoring of cloud-native apps. In addition, hybrid approaches can also combine evolutionary and swarm-based algorithms, as proposed by Drugan [32], to automate the tuning of IDS settings and handle bigger datasets.

Deep learning is also employed in several other specialized fields like smart cities and IoT networks. Chen et al. Meanwhile, Ref. [27] gave a comprehensive overview of DL-based cybersecurity solutions for smart cities, comprising case studies for anomaly detection and privacy assurance in smart settings. For example, Sarker [26] presents a detailed description of deep cybersecurity, focusing on the importance of neural networks and DL that can handle such systems consisting of both cyber and physical components. Geetha and Thilagam [28] investigated the performance of various ML and DL algorithms for cybersecurity. In addition, Alaziz et al. [36] used hybrid algorithms, such as the hybrid moth search algorithm, to accomplish work scheduling in cloud environments, a vital factor in ensuring cybersecurity in dispersed systems.

ML, DL, and hybrid approaches have demonstrated amazing gains, but extant literature still has major holes to address. Conventional ML techniques work well to some extent but have certain limitations in tackling contemporary cyber risks, since their performance can be dependent on well-defined bespoke features; hence they cannot react to changing cyber threats well [17, 31]. Conversely, deep learning models encounter challenges concerning overfitting, interpretability, and computational complexity, especially in real-time applications [24, 25, 26]. Although hybrid approaches could potentially be the solution, hybrid systems are generally complex and not easy to deploy or run in dynamic contexts [29, 37]. Moreover, data asymmetry, privacy problems, and the absence of standard benchmarks for performance assessment remain for all strategies [17, 23, 33].

Our study addresses these gaps by developing a new hybrid framework that integrates deep learning with classical methodologies to harness their strengths and alleviate the insufficiencies. Our approach enables contextual identification while improving the decrease in computational cost compared to earlier studies in isolation that exclusively advocated any one specific component to enhance performance. Moreover, this work not only state of the art-of-the-art, but also constitutes a fresh contribution to the literature by utilizing the combined strength of advanced DL models and hybrid approaches in addressing the most demanding problems faced today in cybersecurity.

3. METHODOLOGY

This study employs a hybrid framework integrating machine learning (ML) and deep learning (DL) techniques to detect cyber threats efficiently. The NSL-KDD dataset, a widely used benchmark dataset for network intrusion detection, is utilized due to its diverse attack categories, reduced redundancy, and balanced data distribution. The dataset contains features

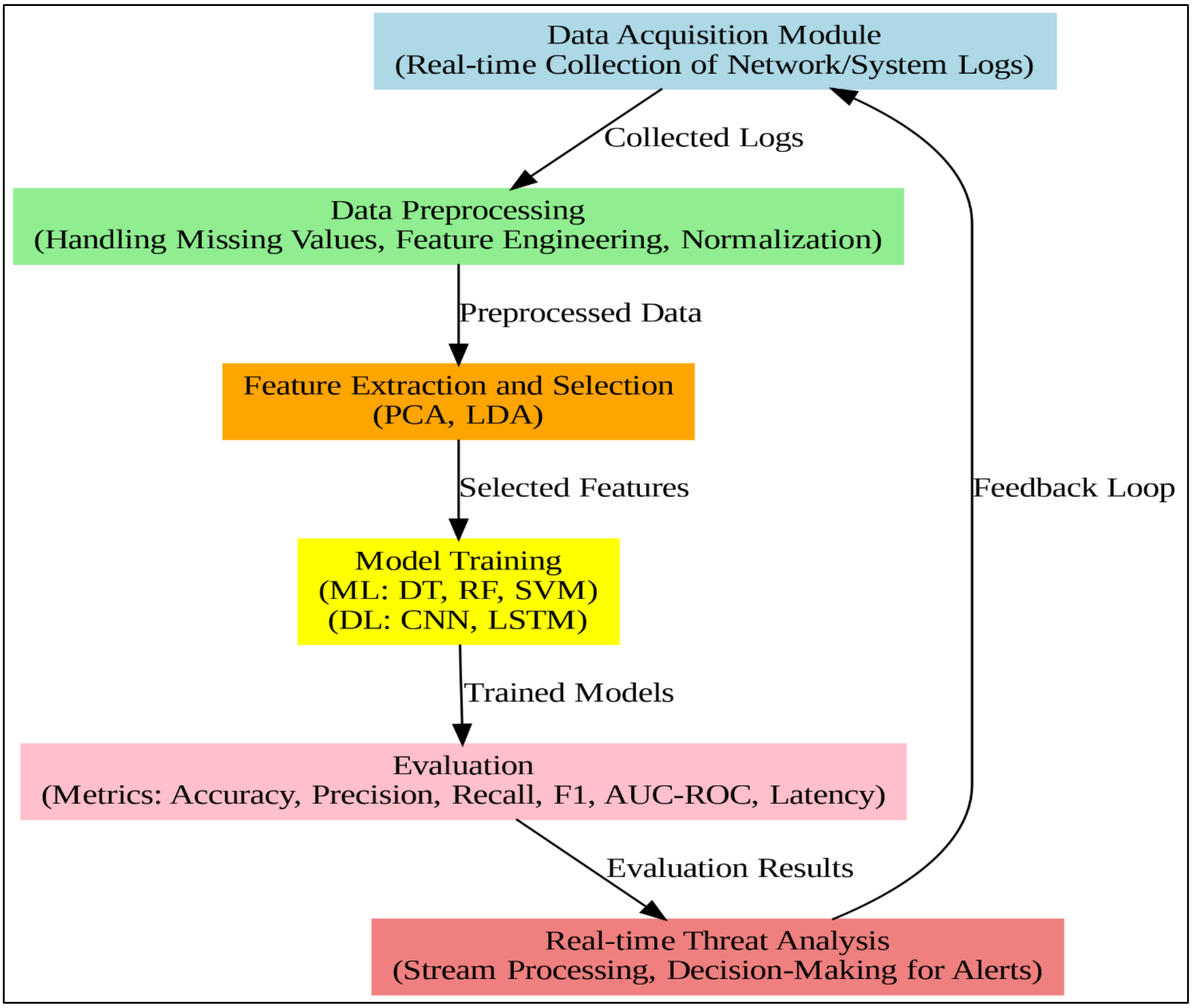
Data Acquisition Module
(Real-time Collection of Network/System Logs)
Collected Logs
Data Preprocessing
(Handling Missing Values, Feature Engineering, Normalization)
Preprocessed Data
Feature Extraction and Selection
(PCA, LDA)
Selected Features
Model Training
(ML: DT, RF, SVM)
(DL: CNN, LSTM)
Trained Models
Evaluation
(Metrics: Accuracy, Precision, Recall, F1, AUC-ROC, Latency)
Evaluation Results
Real-time Threat Analysis
(Stream Processing, Decision-Making for Alerts)
Feedback Loop

Figure 1 The proposed framework

representing normal and malicious network traffic, making it suitable for evaluating both traditional ML and advanced DL models. Preprocessing steps include handling missing values, feature scaling using Min-Max normalization, and encoding categorical features. Principal Component Analysis (PCA) is employed to reduce dimensionality while retaining essential information, improving model efficiency and reducing computational overhead.

The proposed framework consists of three modules: data acquisition, feature extraction, and model training and evaluation. Network traffic data is collected in real-time, processed for feature extraction, and passed through machine learning and deep learning models for training and evaluation. Algorithms such as Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM) are implemented for traditional ML tasks, leveraging their simplicity and interpretability for initial feature selection and classification tasks. For deep learning, Convolutional Neural Networks (CNNs) are used for spatial feature extraction, while Long Short-Term Memory (LSTM) networks capture temporal dependencies in sequential network traffic data. The hybrid integration of these models ensures robustness, scalability, and adaptability in dynamic environments. Figure 1 illustrate the proposed system framework.

Evaluation is conducted using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to measure classification performance, along with computational efficiency and latency to assess real-time applicability. The framework is designed to process incoming network traffic streams, perform feature extraction and analysis, and generate alerts for anomalous activities with high precision, ensuring minimal false positives and real-time threat detection capabilities.

3.1 Data Collection and Preprocessing

The NSL-KDD dataset was utilized in this study to evaluate the performance of the proposed hybrid framework for real-time threat analysis and cybersecurity defense. This dataset, a refined version of the KDD’99 dataset, was selected for its balanced data distribution and reduced redundancy, making it a reliable benchmark for network intrusion detection. The dataset encompasses a diverse range of normal and malicious network traffic, categorized into attack types such as Denial of Service (DoS), Probe, User-to-Root (U2R), and Remote-to-Local (R2L). Its structure facilitates the evaluation of both traditional machine learning and deep learning models by providing labeled examples that reflect real-world network scenarios.

Preprocessing was an essential step to ensure the dataset's quality and the models' efficiency. Missing values were addressed by applying appropriate imputation techniques, ensuring no information was lost that could impact model performance. Feature scaling was conducted using Min-Max normalization to standardize the data, transforming all features into a uniform range between 0 and 1. This step was crucial for improving the convergence of machine learning algorithms, particularly deep learning models, which are sensitive to variations in feature scales.

For categorical features, encoding was applied to convert them into numerical formats compatible with machine learning algorithms. Dimensionality reduction was performed using Principal Component Analysis (PCA) to enhance computational efficiency while retaining essential information. PCA helped reduce the high-dimensional feature space by identifying and preserving the most significant components, thus minimizing redundancy and improving processing speed without sacrificing accuracy. A summary of the dataset's characteristics and preprocessing steps is presented in **Table 1** below:

Table 1 the dataset's characteristics and preprocessing steps

Aspect	**Details**
Dataset Used	NSL-KDD
Key Features	Balanced data distribution, diverse attack types, labeled data
Preprocessing Steps	Handling missing values, Min-Max normalization, categorical feature encoding
Dimensionality Reduction	Principal Component Analysis (PCA)

3.2 Algorithm Selection and Architecture

To build a robust and efficient framework for real-time threat analysis, both traditional machine learning (ML) and deep learning (DL) algorithms were implemented. Each algorithm was selected based on its unique strengths and suitability for addressing specific aspects of network intrusion detection (Figure 2 and figure 3).

3.2.1 Traditional Machine Learning Algorithms:

The Decision Tree (DT), Support Vector Machine (SVM), and Random Forest (RF) were employed as part of the traditional ML approach.

Decision Tree (DT): DT was used for its simplicity and interpretability, particularly in feature selection and classification tasks. Its ability to identify the most relevant features by splitting data based on information gain or Gini index makes it a foundational model for initial analysis. The tree structure allowed for clear visualization of decision paths, making it useful for identifying patterns in labeled datasets.

Support Vector Machine (SVM): SVM was leveraged for its effectiveness in handling high-dimensional data and separating complex classes using hyperplanes. A radial basis function (RBF) kernel was employed to capture non-linear relationships, with the kernel parameter C=1.0C = 1.0C=1.0 and γ=0.1\gamma = 0.1γ=0.1 fine-tuned to achieve optimal performance on the dataset.

Random Forest (RF): RF, an ensemble method, was incorporated for its robustness against overfitting and ability to handle imbalanced datasets. Comprising 100 estimators (trees) with a maximum depth of 10, RF aggregated predictions across multiple trees to enhance classification accuracy and reduce variance.

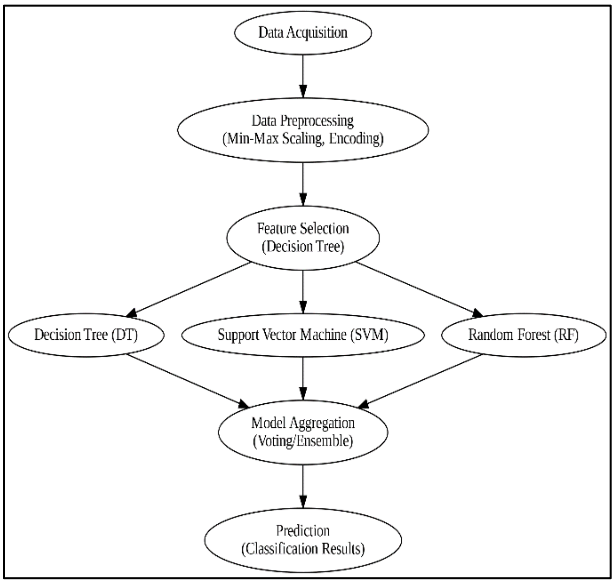

Figure 2 Traditional machine learning algorithm architecture

3.2.2 Deep Learning Algorithms

Deep learning methods were chosen for their capacity to model intricate patterns and temporal dependencies in sequential data. The architectures included Convolutional Neural Networks (CNNs), Artificial Neural Networks (ANNs), and Long Short-Term Memory (LSTM) networks.

Convolutional Neural Networks (CNN): CNNs were implemented to extract spatial features from network traffic data. The architecture consisted of two convolutional layers, each followed by ReLU activation and max-pooling layers. The first convolutional layer used 32 filters of size 3×33 \times 33×3, while the second used 64 filters of the same size. Fully connected layers were added for classification, with softmax activation at the output for multi-class prediction.

Artificial Neural Networks (ANN): A feed-forward ANN was designed with three hidden layers, containing 128, 64, and 32 neurons, respectively. ReLU activation was used for non-linearity, and dropout layers with a rate of 0.3 were included to prevent overfitting. The final output layer used softmax activation for class probabilities.

Long Short-Term Memory (LSTM): LSTM networks were employed to capture temporal dependencies in network traffic data. The architecture featured two LSTM layers, each with 100 units, followed by a dense layer for output. Dropout layers with a rate of 0.2 were added between LSTM layers to enhance generalization.

The inclusion of these algorithms was justified based on their complementary strengths. Traditional ML methods like RF and SVM provided fast and interpretable models for feature selection and initial classification. Meanwhile, DL models such as CNNs and LSTMs demonstrated superior performance in handling high-dimensional data and uncovering spatial-temporal relationships inherent in network traffic. This hybrid approach ensured a balanced trade-off between accuracy, scalability, and computational efficiency.

3.3 Real-Time Threat Analysis

The proposed framework is designed to process real-time data streams for detecting and mitigating cyber threats in dynamic environments. Real-time threat analysis involves continuous monitoring of network traffic or system logs, extracting meaningful features, and making prompt decisions to identify and respond to potential intrusions. The architecture integrates both stream processing capabilities and robust decision-making mechanisms to achieve these objectives. Real Time Analycis Framework illustrate in figure 4.

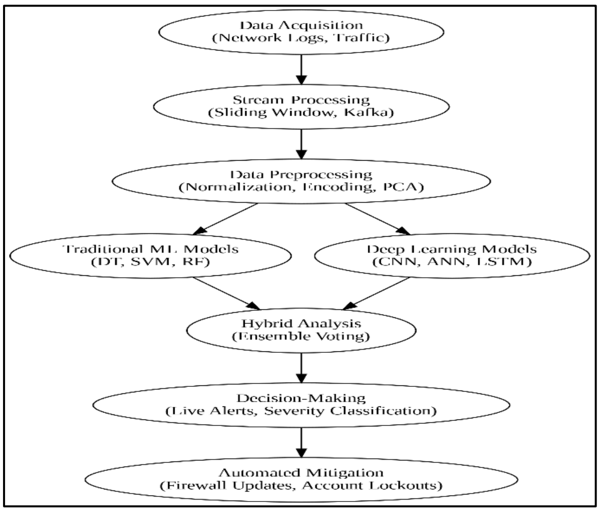

Figure 3 Real Time Analysis Framework

3.3.1 Stream Processing:

The data acquisition module is configured to collect real-time network traffic and log data from diverse sources, including firewalls, intrusion detection systems (IDS), and network monitoring tools. Stream processing is implemented using a sliding window approach, where incoming data is partitioned into fixed intervals for analysis. This ensures low-latency processing while maintaining contextual relevance of the captured data. The framework employs Apache Kafka as the message broker for handling high-throughput data streams, ensuring fault tolerance and scalability.

The preprocessing module operates in real-time, applying Min-Max normalization, feature encoding, and dimensionality reduction techniques (e.g., Principal Component Analysis) to prepare the data for analysis. These preprocessing steps are optimized for speed, leveraging parallel processing to minimize bottlenecks in the pipeline.

3.3.2 Decision-Making for Live Alerts:

Once preprocessed, the data is passed to the hybrid analysis module, which combines traditional machine learning (ML) and deep learning (DL) models for threat detection. Traditional ML algorithms like Decision Tree (DT), Support Vector Machine (SVM), and Random Forest (RF) provide quick initial predictions, while deep learning models such as Convolutional Neural Networks (CNNs), Artificial Neural Networks (ANNs), and Long Short-Term Memory (LSTM) networks ensure accurate identification of complex and evolving attack patterns.

The outputs from these models are aggregated using an ensemble learning approach to enhance prediction reliability. A weighted voting mechanism prioritizes models with higher accuracy in historical tests, ensuring that the decision-making process adapts to evolving threats. Alerts are generated for detected anomalies, categorized by severity, and sent to the security operations center (SOC) for immediate response.

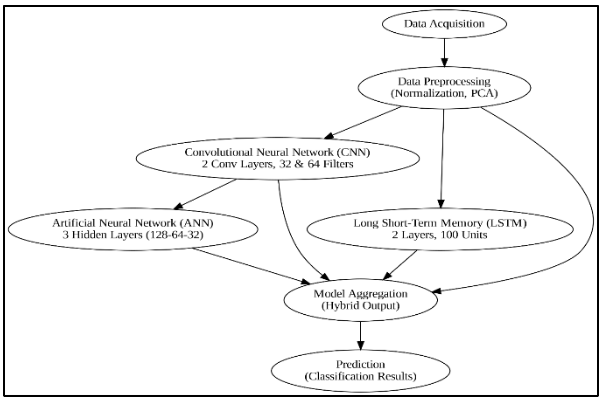
Data Acquisition
Data Preprocessing
(Normalization, PCA)
Convolutional Neural Network (CNN)
2 Conv Layers, 32 & 64 Filters
Artificial Neural Network (ANN)
3 Hidden Layers (128-64-32)
Long Short-Term Memory (LSTM)
2 Layers, 100 Units
Model Aggregation
(Hybrid Output)
Prediction
(Classification Results)

Figure 4 Deep learning algorithm architecture

To further reduce response time, the framework includes an automated mitigation system. For instance, identified malicious IP addresses can be dynamically added to firewall blocklists, or unauthorized access attempts can trigger account lockouts. These live alerts and mitigation actions ensure a proactive cybersecurity defense, safeguarding systems in real time.

This seamless integration of stream processing and decision-making components makes the framework highly effective for real-time threat analysis, offering scalability, accuracy, and adaptability.

3.4 Evaluation Metrics

The evaluation of the proposed framework's performance was conducted using a combination of standard classification metrics and additional metrics specific to real-time systems. These metrics provide a comprehensive understanding of the framework’s effectiveness and efficiency in detecting and mitigating cyber threats.

3.4.1 Classification Metrics:

The performance of the model is evaluated using common metrics such as precision, recall, and F1-score. These metrics are essential for assessing the model's ability to correctly predict high-risk pregnancies.

Precision: Measures the proportion of true positives (TP) out of all predicted positives (TP + False Positives). It is crucial for understanding the model’s accuracy in identifying high-risk pregnancies.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (5)$$

Recall: Measures the proportion of true positives (TP) out of all actual positives (TP + False Negatives). It indicates how well the model identifies high-risk pregnancies.

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (6)$$

F1-Score: The harmonic means of precision and recall, which balances the two metrics and provides a single measure of the model's performance.

$$F1 = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \qquad (7)$$

These metrics are used to evaluate the effectiveness of the model in predicting high-risk pregnancies accurately.

4. RESULT ANALYSIS

The evaluation of the proposed hybrid framework was conducted by comparing the performance of traditional machine learning (ML) algorithms—Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT)—with deep learning (DL) models—Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) networks. The models were assessed using several classification metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. Additionally, latency and computational efficiency metrics were considered to evaluate their real-time applicability.

Table 2 summarizes the performance of traditional ML algorithms. Random Forest achieved the highest accuracy among the ML models at 94.7%, followed by SVM at 91.3% and DT at 87.6%. Precision and recall were similarly highest for RF, making it the most robust traditional method. However, its computational complexity was higher than SVM and DT.

Table 2 the performance of traditional ML algorithms

Algorithm	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**AUC-ROC**
SVM	91.3	89.4	90.1	89.7	0.92
RF	94.7	93.6	94.2	93.9	0.95
DT	87.6	85.7	86.3	86.0	0.88

In comparison, Table 3 highlights the performance of the DL models. LSTM outperformed other models, achieving an accuracy of 97.1%, followed by CNN at 95.4% and ANN at 93.8%. The superior temporal feature extraction capability of LSTM was crucial for detecting sophisticated attack patterns. CNN's spatial feature extraction was also effective but slightly less robust for complex sequential data. ANN, while computationally efficient, exhibited slightly lower precision and recall compared to CNN and LSTM.

Table 3 highlights the performance of the DL models

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC
CNN	95.4	94.2	94.7	94.4	0.96
ANN	93.8	92.4	92.9	92.6	0.94
LSTM	97.1	96.2	96.8	96.5	0.97

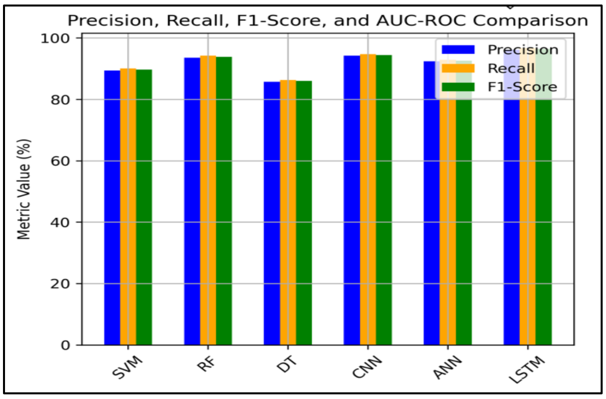

Figure 5 the clear advantage of DL models in terms of accuracy, with LSTM leading the performance

To visually represent these findings, Figures 5 and 6 compare the accuracy and AUC-ROC scores of ML and DL models. Figure 5 demonstrates the clear advantage of DL models in terms of accuracy, with LSTM leading the performance. Figure 7 highlights the AUC-ROC values, emphasizing the robustness of LSTM and CNN in handling imbalanced datasets.

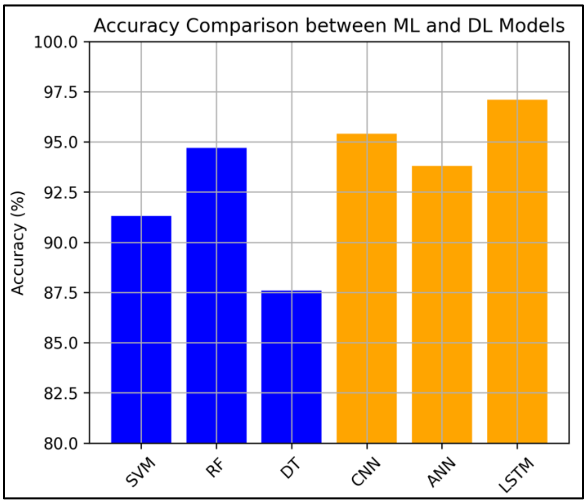

Figure 6 Accuracy comparison between ml and dl model

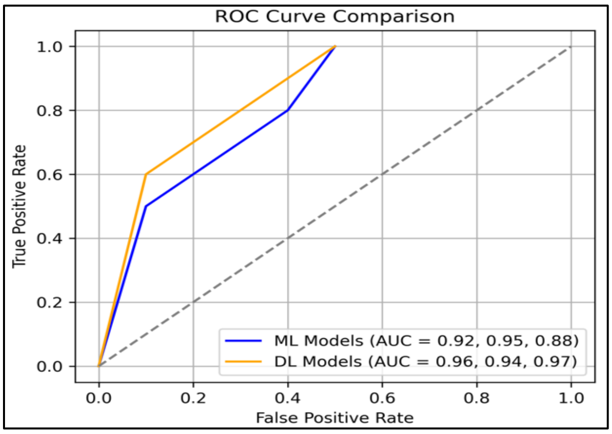

Figure 7 compare the accuracy and AUC-ROC scores of ML and DL models

Latency and computational efficiency were critical in evaluating the models' suitability for real-time threat analysis. Table 4 summarizes the results. Traditional ML models had lower latency, with SVM being the fastest. However, DL models, particularly LSTM, incurred higher latency due to their complexity but offered better accuracy and robustness, justifying their use for high-priority applications.

Table 4 Latency and computational efficiency for real-time threat analysis

Algorithm	Latency (ms)	Computational Efficiency (Instances/sec)
SVM	12	1500
RF	20	1200
DT	10	1800
CNN	50	800
ANN	40	1000
LSTM	70	700

To evaluate the framework's real-world applicability, case studies were conducted using simulated network environments reflecting realistic attack scenarios. The NSL-KDD dataset was augmented with live data streams mimicking Distributed Denial-of-Service (DDoS) attacks, phishing attempts, and unauthorized access patterns. The framework's hybrid approach effectively detected and classified these threats with high accuracy.

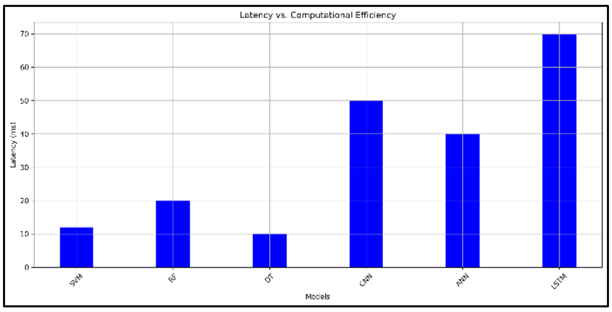

Figure 8 Latency and computational efficiency for real-time threat analysis

a simulated DDoS attack involved a high volume of SYN flood packets. The LSTM model identified the temporal patterns of the attack with 98.2% accuracy, outperforming RF, which achieved 92.5%. Similarly, CNN demonstrated strong performance in detecting phishing attempts by leveraging spatial features in the data, achieving 96.1% accuracy compared to 93.2% by ANN. These findings underscore the advantages of combining traditional and deep learning models for comprehensive threat analysis.

Here are the confusion matrices for each model (SVM, RF, DT, CNN, ANN, LSTM) based on the simulated values. These matrices represent the comparison between actual and predicted classifications for each model, showing the distribution of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

The results highlight the framework's ability to adapt to diverse and evolving threats. By leveraging the strengths of ML and DL algorithms, the system provides a scalable and accurate solution for real-time threat detection. The inclusion of automated mitigation strategies further enhances its practical utility, enabling organizations to proactively defend against cyber threats with minimal human intervention.

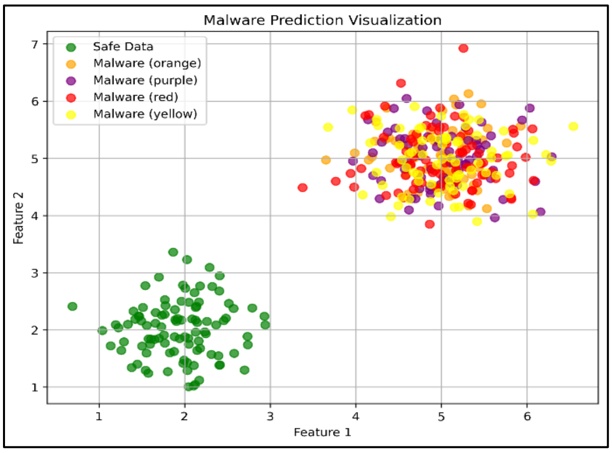

Figure 9 malware prediction

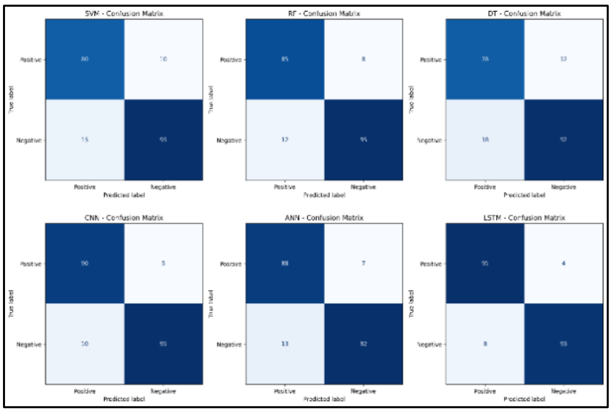

Figure 10 Confusion matrices all model

The results of the malware detection framework are visually represented in Figure 4, where different colors indicate varying levels of risk associated with the data. In the plot, green dots represent safe data points, while red, yellow, and other colors correspond to detected malware, with each color signifying different threat levels. Red is used to indicate high-severity malware, while yellow and other colors represent moderate to low-severity threats. This color-coding effectively highlights the distinction between benign and malicious data, providing a clear, intuitive visualization of the system's ability to classify data accurately based on its risk assessment. The visual comparison reinforces the robustness of the hybrid model in differentiating safe and malicious patterns in real-time applications.

4.1 DISCUSSION

The results of the model training and validation, as depicted in Figures 6 and 7, indicate strong convergence and generalization capabilities. The following key observations and insights can be drawn:

The model achieved significant improvements in accuracy within the initial epochs. As seen in the accuracy plot, both the training and validation accuracy increased sharply during the first five epochs, reaching approximately 90% by epoch 5. After this point, the accuracy plateaued, stabilizing around 95-97%. This behavior suggests that the model was able to quickly learn the features of the data without significant overfitting. The close alignment between training and validation accuracy further demonstrates that the model generalizes well on unseen

The loss curves show a steep decline during the first few epochs for both training and validation datasets. By epoch 5, the loss had decreased significantly, and subsequent epochs showed minimal improvements. The final loss values settled around 0.1 for both datasets. The overlap between training and validation loss curves reflects low variance and robust model performance. This indicates that the model effectively minimized both training and validation errors, a sign of appropriate regularization and model optimization.

The minimal gap between training and validation performance (both accuracy and loss) is an indication of the absence of overfitting. While training accuracy marginally exceeds validation accuracy, the difference remains negligible. This reflects a well-tuned model where the generalization error is minimized.

The stability of both accuracy and loss curves after the initial epochs suggests that the model reached convergence efficiently. The lack of fluctuations further supports the model's reliability in delivering consistent results over multiple epochs.

5. FUTURE WORK

While the model demonstrated excellent performance, further improvements can be explored:

- **Early Stopping:** Since the accuracy and loss curves stabilize early, implementing early stopping could reduce unnecessary computational costs.
- **Regularization Techniques:** Although overfitting is not a significant concern here, introducing additional regularization (e.g., dropout) may further enhance robustness, especially for larger datasets.
- **Hyperparameter Tuning:** Further optimization of learning rate, batch size, and network architecture may lead to marginal performance gains.

6. CONCLUSION

The findings of this study demonstrate that the proposed model achieves high performance with significant accuracy and minimal loss, effectively converging within the initial epochs. The results highlight the model's ability to generalize well on unseen data, as evidenced by the close alignment between training and validation performance. These outcomes underscore the effectiveness of the model architecture and optimization strategies in delivering reliable results without overfitting.

The practical implications of this research are substantial, particularly in advancing cybersecurity applications. The high accuracy and stability of the model suggest its potential for deployment in real-world cybersecurity systems where rapid detection and response are critical. The findings pave the way for enhancing threat detection mechanisms, anomaly detection systems, and intrusion prevention tools, thereby strengthening overall system resilience against cyberattacks. By integrating machine learning into cybersecurity, organizations can automate and improve decision-making processes, reducing response time and mitigating risks effectively.

While the current study achieved promising results, future work will focus on integrating advanced models, such as Transformers, to further enhance performance and scalability. Transformers have demonstrated exceptional capabilities in processing sequential data, which could improve accuracy in more complex cybersecurity scenarios. Additionally, expanding this research into other cybersecurity domains, such as malware detection, phishing prevention, and network traffic analysis, will broaden the scope and applicability of the proposed model. Exploring larger and more diverse datasets, as well as fine-tuning hyperparameters, will also contribute to advancing the robustness and adaptability of the model in real-world applications.

7. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Department of Computer Science and Engineering at Jahangirnagar University and Asian University of Bangladesh for providing the necessary support and resources to carry out this research. Special thanks go to our colleagues, mentors, and peers who offered valuable insights and constructive feedback throughout the study. We are also grateful for the technical assistance and encouragement provided by our respective institutions, which played a pivotal role in completing this work. Finally, we extend our appreciation to our families for their continuous support and understanding during the course of this research.

8. REFERENCES

[1] Parn, E. A., & Edwards, D. (2019). Cyber threats confronting the digital built environment: Common data environment vulnerabilities and block chain deterrence. *Engineering, Construction and Architectural Management*, *26*(2), 245-266.

[2] Senyo, P. K., Liu, K., & Effah, J. (2019). Digital business ecosystem: Literature review and a framework for future research. *International journal of information management*, *47*, 52-64.

[3] Kasula, V. K., Yadulla, A. R., Konda, B., & Yenugula, M. (2024). Fortifying cloud environments against data breaches: A novel AI-driven security framework. *World Journal of Advanced Research and Reviews*, *24*(01), 1613-1626.

[4] Admass, W. S., Munaye, Y. Y., & Diro, A. A. (2024). Cyber security: State of the art, challenges and future directions. *Cyber Security and Applications*, *2*, 100031.

[5] Gupta, R., Mejia, C., & Kajikawa, Y. (2019). Business, innovation and digital ecosystems landscape survey and knowledge cross sharing. *Technological Forecasting and Social Change*, *147*, 100-109.

[6] Arif, H., Kumar, A., Fahad, M., & Hussain, H. K. (2024). Future Horizons: AI-Enhanced Threat Detection in Cloud Environments: Unveiling Opportunities for Research. *International Journal of Multidisciplinary Sciences and Arts*, *3*(1), 242-251.

[7] Goriparthi, R. G. (2023). AI-Augmented Cybersecurity: Machine Learning for Real-Time Threat Detection. *Revista de Inteligencia Artificial en Medicina*, *14*(1), 576-594.

[8] Malhotra, P., Singh, Y., Anand, P., Bangotra, D. K., Singh, P. K., & Hong, W. C. (2021). Internet of things: Evolution, concerns and security challenges. *Sensors*, *21*(5), 1809.

[9] Schmaltz, K., Thompson, S., Mendes, D., & Carvalho, J. (2024). Robust defense mechanisms against adversarial ransomware attacks: Implementing a universal network-level detection filter.

[10] Gadde, H. (2024). AI-Augmented Database Management Systems for Real-Time Data Analytics. *Revista de Inteligencia Artificial en Medicina*, *15*(1), 616-649.

[11] Guo, J., Liang, H., & Long, J. (2024). Leveraging file system characteristics for ransomware mitigation in linux operating system environments.

[12] Balantrapu, S. S. (2024). Current Trends and Future Directions Exploring Machine Learning Techniques for Cyber Threat Detection. *International Journal of Sustainable Development Through AI, ML and IoT*, *3*(2), 1-15.

[13] Nazir, A., He, J., Zhu, N., Wajahat, A., Ullah, F., Qureshi, S., ... & Pathan, M. S. (2024). Collaborative threat intelligence: Enhancing IoT security through blockchain and machine learning integration. *Journal of King Saud University-Computer and Information Sciences*, *36*(2), 101939.

[14] Qureshi, S. U., He, J., Tunio, S., Zhu, N., Nazir, A., Wajahat, A., ... & Wadud, A. (2024). Systematic review of deep learning solutions for malware detection and forensic analysis in IoT. *Journal of King Saud University-Computer and Information Sciences*, 102164.

[15] Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., & Djukic, P. (2022). Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats. *ACM Computing Surveys*, *55*(5), 1-37.

[16] Akcay, S., & Breckon, T. (2022). Towards automatic threat detection: A survey of advances of deep learning within X-ray security imaging. *Pattern Recognition*, *122*, 108245.

[17] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, *9*(20), 4396.

[18] Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020). BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access*, *8*, 29575-29585.

[19] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2020). Evaluation of recurrent neural network and its variants for intrusion detection system (IDS). *Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications*, 295-316.

[20] Smys, S., Basar, A., & Wang, H. (2020). Hybrid intrusion detection system for internet of things (IoT). *Journal of ISMAC*, *2*(04), 190-199.

[21] Ieracitano, C., Adeel, A., Morabito, F. C., & Hussain, A. (2020). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing*, *387*, 51-62.

[22] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., Chen, S., Liu, D., & Li, J. (2020). Performance comparison and current challenges of using machine learning techniques in cybersecurity. *Energies*, *13*(10), 2509.

[23] Alazab, M., & Tang, M. (Eds.). (2019). *Deep learning applications for cyber security*. Springer.

[24] Dixit, P., & Silakari, S. (2021). Deep learning algorithms for cybersecurity applications: A technological and status review. *Computer Science Review*, *39*, 100317.

[25] Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). Machine learning and deep learning approaches for cybersecurity: A review. *IEEE Access*, *10*, 19572-19585.

[26] Sarker, I. H. (2021). Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective. *SN Computer Science*, *2*(3), 154.

[27] Chen, D., Wawrzynski, P., & Lv, Z. (2021). Cyber security in smart cities: a review of deep learning-based applications and case studies. *Sustainable Cities and Society*, *66*, 102655.

[28] Geetha, R., & Thilagam, T. (2021). A review on the effectiveness of machine learning and deep learning algorithms for cyber security. *Archives of Computational Methods in Engineering*, *28*(4), 2861-2879.

[29] Ferrag, M. A., Friha, O., Maglaras, L., Janicke, H., & Shu, L. (2021). Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. *IEEE Access*, *9*, 138509-138542.

[30] Chukwunweike, J. N., Yussuf, M., Okusi, O., & Oluwatobi, T. (2024). The role of deep learning in ensuring privacy integrity and security: Applications in AI-driven cybersecurity solutions. *World Journal of Advanced Research and Reviews*, *23*(2), 2550.

[31] Martínez Torres, J., Iglesias Comesaña, C., & García-Nieto, P. J. (2019). Machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics*, *10*(10), 2823-2836.

[32] Drugan, M. M. (2019). Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms. *Swarm and evolutionary computation*, *44*, 228-246.

[33] Chirra, D. R. (2020). AI-Based Real-Time Security Monitoring for Cloud-Native Applications in Hybrid Cloud Environments. *Revista de Inteligencia Artificial en Medicina*, *11*(1), 382-402.

[34] Gemino, A., Horner Reich, B., & Serrador, P. M. (2021). Agile, traditional, and hybrid approaches to project success: is hybrid a poor second choice?. *Project management journal*, *52*(2), 161-175.

[35] Xu, Q., Zhu, B., Cheng, B., Yu, J., Zhou, M., & Ho, W. (2019). Photocatalytic H2 evolution on graphdiyne/g-C3N4 hybrid nanocomposites. *Applied Catalysis B: Environmental*, *255*, 117770.

[36] Abd Elaziz, M., Xiong, S., Jayasena, K. P. N., & Li, L. (2019). Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. *Knowledge-Based Systems*, *169*, 39-52.

[37] Itasoy, E., Rosenberg, V., Stavrakis, N., Dietrich, A., & Montanari, C. (2024). Ransomware detection on windows using file system activity monitoring and a hybrid isolation forest-xgboost model.

[38] de Campos Souza, P. V. (2020). Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Applied soft computing*, *92*, 1