Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Adaptive Cybersecurity Framework for Distributed Computer Systems using Advanced Machine Learning Techniques

Goldi Soni
Assistant professor, Amity University Chhattisgarh, Raipur, India,

Divyanshu Dewangan
Student MCA3rd Amity University Chhattisgarh, Raipur, India,

Rahul Verma
Student MCA3rd Amity University Chhattisgarh, Raipur, India,

Gulshan Verma
Student MCA3rd Amity University Chhattisgarh, Raipur, India,
https://orcid.org/0009-0000-6615-6815

Alka Dewangan
Student MCA3rd Amity University Chhattisgarh, Raipur, India,

Aryan Verma
Student MCA3rd Amity University Chhattisgarh, Raipur, India,

Pinky Vishwankarma
Student MCA3rd Amity University Chhattisgarh, Raipur, India,

Abstract - With the rapid advancement of distributed computing systems, the complexity of cybersecurity approaches has risen due to the numerous elements of interconnected nodes, communication paths, systems, and resources assuming numerous stages for cyberattacks. Traditional approaches to security are mostly rule-based and signature-based ones that will likely face challenges in counteracting new attacks. The article describes an innovative approach referred to as the Machine Learning Cybersecurity Framework. Through this framework, it is expected to observe network traffic, system logs, authentication information, system activities, and access information on distributed nodes. The gathered information is being processed and comes through the stage of preprocessing, normalization, and conversion into peculiar security features. The framework's approach has resulted in the training of a machine learning model that is used in the identification of normal, suspicious, and dangerous activities based on the received features. An incident risk score is estimated. The system compares the score against a set threshold and executes necessary measures, such as generating alerts, blocking communication, and isolating nodes.The developed approach is studied against Random Forest, SVM, XGBoost, KNN, Decision Tree, Naïve Bayes, Logistic Regression, ANN, CNN, and LSTM. The results of the experiment yielded an accuracy rate of 98.74%, a precision equal to 98.52%, a recall rate equal to 98.31%, an F1-score of 98.41%, and a specificity of 99.08%, resulting in an FPR of 0.92%. The threshold shows the reliability of the developed approach.

Keywords - Adaptive Security, Artificial Intelligence, Cybersecurity, Distributed Systems, Intrusion Detection, Machine Learning, Network Security, Risk Assessment, Threat Detection, Vulnerability Detection

1. INTRODUCTION

The emergence of advanced computer networks has changed how people manage and transmit information in organizations. In the latest world, corporations depend on various distributed systems such as the various cloud computing frameworks, IoT systems, edge computing systems, distributed databases, remote servers, and the interconnectivity of enterprise applications due to the benefits of being flexible, scalable, efficient, and highly available. [2], [3], [11] Nevertheless, the distributed nature makes the cybersecurity environment broader and much more complex than centralized ones. In contrast to traditional centralized network tools, distributed systems include different channels of communication and objects that communicate with each other, thus creating additional points for cyber threats. Hackers can successfully take advantage of the vulnerabilities of the endpoints of the new system or the channels.[3], [13], [15]

Conventional techniques in the field of cybersecurity are mainly based on specific sets of rules, signatures, access-control policies, and security techniques that have been set up manually in advance. While these techniques are quite effective against known threats, they often face challenges when it comes to identifying new threats that are not yet available in any threat information database or in any security system. Current cyber threats can change their behavior, emulate legitimate network operations, and create a huge amount of traffic in order to evade detection. Therefore, there is a need for security solutions that can carry out continuous analysis of how the system behaves in order to detect any anomalies. Machine learning (ML) has

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

appeared on the scene as an answer to the existing requirement since it can be used for analyzing a huge amount of data in order to identify any suspicious behavior.[3], [4], [19], [20].

A cybersecurity framework powered by machine learning has the capability to scrutinize network traffic, system logs, records of authentication, user behavior, and patterns of access to resources for the purpose of identifying deviations from typical system behavior. Classification algorithms help distinguish between legitimate and malicious activity, while clustering and anomaly detection algorithms allow identifying previously unseen behavior. Furthermore, machine learning enables real-time monitoring of security issues with regular updating of understanding of network conditions. Hence, its integration with security in distributed systems can help increase the capabilities of organizations with respect to the detection of attacks in their early stages and the management of security events.[2], [19], [20]

Nevertheless, the use of machine learning in distributed cybersecurity environments poses several problems. Distributed systems are always producing heterogeneous and high-dimensional data from various channels at all times. The security-related data may also be imbalanced since normal operations happen much more frequently than attacks. Additionally, there may be privacy concerns that inhibit centralized collection of sensitive data from different devices or organizational domains. The performance of the models may also diminish due to changes in network behavior. Consequently, an effective cybersecurity scheme should take into account not only detection accuracy, but also scalability, adaptability, privacy, communication efficiency, and timely response.

1.1. Current Developments

Recent advancements in cybersecurity imply a definite trend toward the emergence of automated and intelligent threat-detection systems. Today, machine learning and deep learning methods are widely used for intrusion detection, malware identification, phishing identification, abnormal user behavior analysis, botnet identification, and abnormal network activity detection. The traditional machine learning methods, represented by Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), k-Nearest Neighbor (k-NN), and the Naïve Bayes method, are still very common approaches used in the processing of cybersecurity information because of their ability to recognize types of network activities by learning from the past.[3], [4], [11], [19]

Deep learning has enhanced cybersecurity capabilities by allowing the extraction of complex features from big security data. Neural networks, convolutional neural networks, recurrent networks, and other deep architectures can discover complicated relationships between network traffic and sequential system events. Such technologies become extremely helpful in the area of cybersecurity when the data sets contain a huge number of features and constantly changing attack patterns.

One more crucial aspect is distributed and federated learning in terms of cybersecurity. Distributed learning allows the nodes to take part in the training of the models without having all the security data available on a single server. Federated learning is important for issues of privacy, since it allows organizations or local devices to conduct the training based on their local data, which makes it possible to avoid sharing sensitive security data.[8]–[10], [18]

Security at the edge continues to gain importance. Processing security information closer to where it is sourced minimizes communication time and improves response time. Nevertheless, edge technology generally has limited computational and storage capabilities, thus requiring efficient and lightweight machine learning algorithms. The future of cybersecurity systems must integrate intelligent detection, distributed processing, fast response, and privacy-preserving technology.

1.2 Principle

The main goal of a cybersecurity framework that uses machine learning is to offer constant, appropriate, and intelligent protection to an entire distributed computer system. This system should be able to detect attacks by understanding what kind of behavior can be called normal and which are abnormal. Furthermore, rather than relying only on attack signatures, this framework uses information analysis and does not depend entirely on known threats.Top of FormBottom of Form Top of Form Bottom of Form
The framework is based on several main principles. First of all, there needs to be constant monitoring in order to keep an eye on network traffic, user actions, system occurrences, and resource access patterns. Secondly, it is necessary to preprocess the data along with matching relevant features. Thirdly, the framework depends on intelligent classification in order to detect unusual behavior. Lastly, the framework should enable making decisions in an effective manner, which implies ranking threats according to their seriousness.

Adaptability is another vital aspect of the threat model. With the continuous change in cyber threats, the security model must also adapt to changes in the way the network behaves and attacks occur. Flexibility is one of the main concerns of the model, which must also allow for scaling up by adding

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

new devices and nodes to the network. It is also very important to maintain the privacy of the communication between the distributed nodes whenever security-related information or the model itself is updated.[2], [3], [19]

1.3 Solutions Proposed

In order to overcome the shortcomings of current methodologies of cyber security, a machine-learning-oriented cybersecurity framework for safeguarding all-inclusive computer systems is presented in this research work. This framework provides a unified structure comprising the process of data gathering and preprocessing, feature selection, recognition of the threat based on machine learning, risk assessment, and provision of security measures.

First of all, security data is collected at various distributed locations, which includes network traffic, system logs, user activities, authentication incidents, and records of access to many important resources. After that, the gathered data is subjected to a preprocessing phase, during which noise, extra features, and incomplete records are eliminated from the data.

The learning algorithm is now receiving information that has been analyzed so that it can determine whether or not the behavior in question is suspicious or legitimate. The model uses its ability to learn and categorize behaviors into security classifications. When suspicious behavior has been revealed, information about the risks posed by the activity is analyzed, depending on the complexity of the problem. For example, in some cases, when a simple issue is taking place, the system is only used to generate alarms. In more complicated cases it can be allowed to cut accesses and even block viruses provided that the system receives the proper permission.

One more benefit in terms of the usage of the proposed solution is that it can enable distributed deployment, which, in turn, means that security evaluation can take place closer to various elements of the system. Thus, this suggests that there is no need to rely on one centralized point of security, resulting in better scalability of the system.[2] [7], [19], [20]Top of Form Bottom of Form Top of Form Bottom of Form

1.4 Main Contributions

The main contributions of this study are as follows:

1. A cybersecurity model based on machine learning is suggested for safeguarding distributed computer systems from various cyber attacks.
2. An exceptional approach to threat detection is created for detecting irregular and harmful behavior based on data from distributed systems and network.
3. Data preprocessing and unique strategies for selection of features are used to enhance the performance of security data, while minimizing excessive computational load.
4. Adaptive technique for machine learning analysis is aimed at enhancing detection efficiency against new patterns of attacks that are not seen before.
5. Distributed security monitoring is applied to ensure extensibility of security measures with regard to multiple connected devices and network nodes.
6. Rea l-time threat detection and response systems are incorporated to minimize the length of time between an attack detection and security measures taken.
7. Risk-based security decisions take place whereby identified risks are ranked based on their level of impact and severity.[2], [8], [13].
8. An approach known as privacy-aware distributed learning has been used to minimize the transmission of sensitive data among participating nodes.
9. The evaluation of performance is carried out using different measures including accuracy, precision, recall,

2. LITERATURE REVIEW

Explanation of Popular Machine Learning Methods for Cybersecurity

The application of machine learning techniques in the field of cybersecurity allows for the detection of malicious activities, recognition of intrusion in networks, and classification of attacks in distributed computing systems. The algorithms chosen depend on the specific requirements. Following are the top ten methods applicable in building Cybersecurity Framework for the Protection of Distributed Computing Systems Using ML.

2.1 Random Forest (RF)

Random Forest is a type of machine learning algorithm used in ensemble learning that involves the combination of multiple decision tree models to create better outcomes. In cybersecurity, RF can be used to classify whether the network traffic is normal or malicious and which specific attack is taking place. The tool is particularly effective since it has the ability to deal with high volume of data containing many variables and includes measures of over-fitting that would not be possible with a single tree. Another plus of RF is that it can provide with information about the importance of various features helping understand which aspects of the network can be used for detection of the attack.[11], [19].

2.2 Support Vector Machine (SVM)

The Support Vector Machine is a supervised machine

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

learning technique used primarily for classification. SVM tries to determine the best boundary, or hyperplane, which separates multiple categories of data. In cybersecurity, SVM is thus able to tell the difference between safe and harmful network connection attempts. This method works best with multiple features consisting of various kernel functions to manage the nonlinear relation between input variables, though it is expensive computationally in cases of handling large data.[3], [4], [19]

2.3 XGBoost

XGBoost refers to optimized gradient-boosting algorithm that is built on constructing decision trees whose efficacy relies on correcting the mistakes of already constructed trees in the building of the model. The usefulness of this algorithm is beneficial in its application in making predictions in the realm of cybersecurity, as it models complex relations among detection features. The algorithm can be utilized in intrusion detection and malware classification, as well as in attack prediction. .

2.4 K-Nearest Neighbors (KNN)

KNN classifies an observation by looking at its neighbors' classes. When used in cybersecurity, KNN compares the current behavior of the network to previously known behaviors of the network and returns a conclusion about the nature of the activity. It is easy to use since there is no complex training of the model but predictions can be time-consuming when the training dataset is large.

2.5 Decision Tree (DT)

A decision tree is a method of making categorization decisions using a tree-like model based on a set of input variables. Internal nodes imply decisions while the leaves show the classification results. Decision trees can be used in cybersecurity to classify network traffic in good or malicious behavior. The key benefit of this approach is that it offers good interpretability meaning that security professionals can easily understand the model’s decisions. On the other hand, a decision tree can also be influenced by overfitting.[11], [19]

2.6 Naïve Bayes (NB)

Naïve Bayes is a classification algorithm based on Bayes' theorem that relies on the conditional independence of the features being used to predict the target variable. This means that, no matter how false this assumption may be, Naïve Bayes still allows for very fast and effective classification. In cybersecurity, this can be applied to intrusion detection, spam detection, classification of malicious events, and detection of suspicious activities due to the low amount of computational resources required.

2.7 Logistic Regression (LR)

Logistic Regression is a statistical technique used in machine learning that is mainly used for the classification of two classes. It computes the probability of an observation belonging to a certain class. In Cybersecurity, it can be used to categorize network events into classes such as legitimate and malicious. Logistic Regression is simple to compute and easy to interpret, but since it normally uses a linear decision boundary, it may not deal with more sophisticated attacks and patterns as good as modern methods that include advanced ensemble or deep learning.

2.8 Artificial Neural Network (ANN)

ANN is a computing system based on the working of biological neural networks. An ANN comprises a series of layers of neurons that work together and learn the relationship that exists between input features and output labels. In the cyber security space, ANN tools are able to analyze data like network traffic, system logs, user behavior, etc. and help to identify complex attack patterns that are not easy to identify with the traditional statistical models. However, ANN tools require sufficient data and enough computing power too.[3], [20]

2.9 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) were designed specifically for image processing, but they can also be adapted for use in structured and transformed cybersecurity data. CNNs work by performing convolutions to learn the important features in the input data automatically. CNN scan be used in many cybersecurity applications including network traffic classification, malware detection, intrusion detection, and feature extraction. Although CNNs require a large amount of computational power, they are helpful in performing automatic feature learning..[5], [20]

2.10 Long Short-Term Memory (LSTM)

Long Short-Term Memory is one of the advanced classes of recurrent neural networks that are created to discover the relationship in serialized data. Cybersecurity data has time-related characteristics, including network connections, the events of logging in, and performing actions in the system. LSTM is able to capture this ability to learn from the time-dependent sequences and to recognize the events that could be a sign of an attack. Thus, it helps to establish intrusion detection and to analyze behaviors in distributed systems. At the same time, LSTMs require longer learning time and more equipment capacity compared to classical algorithms.

In general, all these techniques represent diverse capacities. Random Forest, XGBoost, and Decision Tree are successful in making structured classifications, while SVM, KNN, NB,

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

and LR may represent relatively simple approaches. On the other hand, ANN, CNN, and LSTM methods can enable advanced learning capabilities in complex time-related security patterns.

3.PROPOSED METHODOLOGY:

The Machine-Learning-Based Cybersecurity Framework for Distributed Computer Systems being proposed is likely to provide a new and efficient way to safeguard distributed computing systems from the new cyber insecurity issues. The framework is able to analyze information related to security breaches and check whether suspicious events are happening in the distributed computing environment. The process starts with the initialization of the distributed environment consisting of the nodes, the sources of security data, and the parameters of the machine learning model. Information about things such as system logs, network traffic, authenticating logs, user activity, connection patterns, and resource access is collected from various distributed nodes. In case the security database contains some duplicate, missing, irrelevant, or problematic data, some data pre-processing must be performed to make the data less unreliable. Feature selection is implemented to determine the significant features and to eliminate unnecessary data and complexity of calculations. Subsequently, the dataset is partitioned into the training, validation, and testing sets in order to ensure a more reliable model creation and assessment. The selected features and attack labels are used to teach the ML model about normal behavior and attacks for the further procedure. After that, the model is verified based on the validation data, which make it possible to determine whether the model is able to work with new security events. If required, adjustments can be made to the parameters of the model in order to improve its performance. Once the acceptable level of validation performance is achieved, the model is applied to analyze incoming records on an ongoing basis. Each new record is analyzed by the model that evaluates the situation and predicts the probability of the attack.The risk score indicates the possible seriousness of the incident, and it is compared with a predetermined security threshold. If incidents are within the permissible risk level, they can be handled normally; while if they go above this threshold, they will be considered dangerous. The system is capable of categorizing the incidents as normal, suspicious or malicious, based on the anticipated consequence of the incident.[2], [13], [19].

In the case of triggering of severely dangerous events, the system generates an alert for either the security administrator or monitoring system. With the highly risky events, the system may take necessary steps such as closing the communication channel, isolating the suspected participants, or increasing the monitoring of the chosen events, etc. Isolation of nodes is extremely important in distributed systems, as it may help to stop the malicious event spread by one of the connected elements. All events, alerts, responses, as well as the results of the response, are recorded in the security monitoring system. These records are useful for further events analysis, security audits, threats investigation, as well as for model improvement. Once the cybersecurity framework is implemented, its effectiveness is assessed based on the results of tests that are performed on specific data. The main indicators of effectiveness include precision, recall, F1 score, detection rate, and false positive rate. Precision is the measure of how reliable the detected threats are, whereas recall is used to measure the effectiveness of the framework in detecting genuine attacks. While F1 score indicates the balance between reliability and effectiveness of detection, and detection rate indicates the percentage of successful attacks that were detected, false positive rate is an indicator of how many legitimate operations were mistaken for attacks. After a detailed examination of these indicators, the framework can conclude whether there is a need to modify and update the model.

New effective security practices can be implemented in the model's training so that this model can adapt itself to new network demands. Continuous learning makes the AI-based framework suitable for use in unconventional distributed systems.In conclusion, the suggested approach combines all the data gathering, data preprocessing, feature extraction, feature selection, machine learning-based prediction, risk assessment, threat classification, alert generation, automated incident response, performance assessment, and model updating processes into one framework. The idea is to create a flexible comprehensive approach to protection of networked computer systems which minimizes the reliance on rigid security rules. The method will enable early discovery of potential attacks, improve reaction capability, mitigate possible harm to the system, and enhance the security of interconnected computing facilities.[8], [9], [18]

3.1 Proposed Algorithm 1: Machine-Learning-Based Cybersecurity Framework for Distributed Computer Systems

Step 1: Initialize the distributed cybersecurity environment with nodes, security data, and learning parameters.

$$D = \{(x_i, y_i)\}_{i=1}^{N} \quad (1)$$

$$x_i \in \mathbb{R}^d \quad (2)$$

$$y_i \in \{0, 1, \ldots, C-1\} \quad (3)$$

Step 2: Collect network traffic, system logs, authentication records, and user activity from

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

distributed nodes.

$$D = \bigcup_{k=1}^{K} D_k \quad (4)$$

Step 3: Use the collected information to delete the same, broken, or fake security logs.

Step 4: Normalize the numerical parameters so that inputs with different scales can be processed in the same way.

Step 5: Devise interesting features for security analysis from network packets, connection behavior, access patterns, and system events.

$$X' = \{x_i' \mid x_i \in X\} \quad (5)$$

$$x_i' = f(x_i) \quad (6)$$

Step 6: Carry out the process of feature selection to ensure that relevant features are retained for the process of threat detection.

$$F^* = arg\,max_{F \subseteq F} F) \quad (7)$$

$$S(F) = \alpha A(F) - \beta R(F)$$

$$A(F) = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

Step 7: The dataset has to be split into different subsets, which includes training dataset, validation dataset and testing dataset.

Step 8: Use the data that has been procured through previous steps to train a model against the new security features retrieved.

$$\theta^* = arg\,min_{\theta} D_{train};\theta) \quad (9)$$

$$\hat{y} = g_{\theta^*}(x_i) \quad (10)$$

N

Step 9: Using the model which has been trained, different validation samples will have to be analyzed and accordingly changes will be made in case changes are required.

Step 10: Use the trained model to analyze the relevant incoming events relating to the distributed system.

$$p_i = P(y_i \mid x_i;\theta^*) \quad (11)$$

$$r_i = max_y P(y \mid x_i;\theta^*) \quad (12)$$

Step 11: The different calculated risk values for the types of activities.

$$R_i = w_1 p_i + w_2 s_i + w_3 e_i \; 0 \leq R_i \leq 1$$

$$T_i = \mathbb{I}(R_i \geq \tau) \quad (13)$$

Step 12: The calculated risk value is compared against the fixed risk threshold.

Step 13: Based on the activities and the risk factor, the activities will be classified as normal, suspicious or malicious.

Step 14: In case the security level does exceed the acceptable limit, alerts are to be generated.

$$A_i = \mathbb{I}(R_i \geq \tau)$$

Step 15: Appropriate response against the risky activities will then be implemented.

$$C_i = h(R_i)$$

$$R_{resp} = \{block, isolate, alert, monitor\}$$

$$a_i = arg\,max_{a \in R_{resp}} a, R_i) \quad (14)$$

Step 16: Isolate or restrict suspicious nodes when the nodes have been engaged in activities that indicate serious threats to security.

0

Step 17: Keep documentation of the detected events, alerts, responses, and the outcomes of the alarmed situation in the system for security monitoring.

Step 18: Assess the performance of the detection process by using test data.

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

Step 19: Determine the detection performance and check if it is necessary to update the model.

$$F1 = \frac{2PR}{P + R} \quad (17)$$

$$DR = \frac{TP}{TP + FN} \quad (18)$$

$$FPR = \frac{FP}{FP + TN} \quad (19)$$

Step 20: Update the model with new verified information about security and perform monitoring continually.

3.1Notations Used in Algorithm

This algorithm employs multiple symbols to express security information in a distributed manner, parameters of the model, predictions, and performance Scores. D stands for the complete security database while D_k indicates the database acquired from k-th distributed node. N is the count of observations and d stands for the count of features. x_i is

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

used for an input record while y_i stands for its security class. X^' stands for the features obtained while F^* indicates the selected optimal feature subset. S(F) denotes the feature selection score. TP, TN, FPand FN stands for true positive, true negative, false positive and false negative values respectively. θ symbolizes machine learning model parameters while θ^* refers to the optimal parameters. L represents the training loss. (y⊥^)_ indicates predicted Class. p_i stands for the forecasted possibility of occurrence of an event in security Class. R_i means evaluated security risk and τ signifies the predetermined risk level. A_i refers to the alert issued for security. C_i stands for assessed type of response. P, R, F1, DR and FPR refer to precision, recall, F1-score, detection rate and false positive rate.

3.2Explanation of Algorithm

Algorithm 1 is an innovative security method that can prevent distributed computer systems from cyber attacks. The method starts by gathering information on security from the various distributed nodes. The information includes the network activity, logs of the operations done in the system, authenticity records, and user activity. The collected data is refined and normalized so as to achieve better quality and consistency of the data. At last, relevant features are chosen in order to eliminate unnecessary details and make the model operate more efficiently.

The chosen features are used to train a model through machine learning. The model recognizes the patterns of normal activities and illegal actions with the help of verification based on other data. When the model is working, it processes each system event.

If the risk becomes greater than that considered acceptable in advance, then an alert is triggered and the appropriate response is selected. High-risk nodes can be quarantined or their functionality restricted to prevent any more damage from being done. All events detected, as well as responses to them, are recorded so that they could be analyzed later. The model is assessed with the use of precision, recall, F1-score, detection rate and false positive rate. Newly provided security information can be used in further training.

4. RESULT

The suggested Machine-Learning-Based Cybersecurity Framework delivers an intelligent, adaptable, and all-encompassing method for securing distributed computer systems from the fluctuating cyber dangers. The system is meant to monitor distributed network nodes unceasingly, accumulate security-related information, discover abnormal behaviors, measure security dangers, and take the required protective measures. The procedure starts with gathering various types of security data such as network traffic, system logs, authentication data, information about users' actions, connection behavior, and access to resources of different distributed nodes. Since the gathered data might consist of noise, duplicated records, missing values, or contradictory information, preprocessing takes place in order to improve data quality and reliability. Numerical characteristics are standardized so that the processing of numerical data is uniform, and after that meaningful security parameters are extracted from network packets, communication style, users' behavior, authentication actions, and system activities. Then the process of feature selection occurs, where the most informative features are selected while eliminating irrelevant data and reducing the complexity of computations. Finally, the processed data is divided into training, validation, and testing datasets so that machine-learning models can be developed reliably and assessed.At the beginning of the learning stage, some features and associated security labels are given to the model to help it discover any critical patterns related to both normal and harmful activities. Once the training is over, the next step involves evaluating its performance on new data to discover whether the model can solve upcoming security problems. Deployment means that an incoming stream of network and system events will be analyzed by the trained model. The model provides each event with its predicted security status and level of confidence. Next, it calculates a risk score based on the consequences of an event occurring. Following that, the risk score is compared with some preestablished thresholds whereupon decisions about whether an event falls within the category of normal, suspicious, or harmful activities are made. If an event exceeds the set threshold, the system generates an alert and conducts some actions depending on the level of risk that needs to be mitigated.In distributed computer systems, the isolation of high-risk nodes is crucial because it allows to reduce the spread of malicious operations to other interconnected entities. All the events, alerts, responses, and results obtained are recorded to conduct security analysis and improve models in the future. The efficiency of the proposed framework was tested against ten machine learning methods, namely Random Forest, Support Vector Machine, XGBoost, K-Nearest Neighbors, Decision Tree, Naïve Bayes, Logistic Regression, Artificial Neural Network, Convolutional Neural Network, and Long Short-Term Memory. The results of the comparison are given in Table 1 and Figure 1 show that the proposed method’s classification successes exceed those reached via other methods employed. The accuracy of the proposed method equals to 98.74%, which is bigger than all the previously existing methods.The model attained an accuracy level of 98.52%. This means that the high portion of detected activities that are classified as harmful is composed

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

of correctly identified events. When it comes to the recall (sensitivity), the value raised up to 98.31%. This means that the model is efficient in the discovery of real attacks on the IT system. The F1-score of 98.41% shows that the model maintains a proper balance between precision and recall.[3], [4], [19], [20]

Additionally, the framework reached a specificity of 99.08%, which proves its capability of correcto distinguishing between real operations in the system. The false positive rate is rather low -0.92%. The findings indicate that the model shows less propensity to misinterpret regular actions as harmful behavior in the system. Among the existing methods, LSTM has demonstrated high results, followed by CNN and XGBoost, while Decision Tree and Naive Bayes have shown relatively poor results. The numerical representation presented in the Figure 1 demonstrates the effectiveness of the investigated framework in terms of accuracy, precision, recall, F1-score, and specificity.The improvement is due to the utilization of the integrated framework, which includes the processes of data preprocessing, relevant feature selection, machine learning-based classification, risk assessment, automated response, and continuous monitoring. In contrast to the systems that mainly deal with static classification, the integrated framework allows for continuous security evaluation and modification of security models based on the latest verification results of security information. This makes it possible to modify the framework in compliance with changes in networks and appearance of new cyber-attack methods.

The improved rate of false-positive results can also be considered as a benefit since too many false alarms would lead to overload of security personnel and create hurdles in performing security actions. The intention is to specify the advanced performance of the integrated framework, which has an opportunity to ensure better results in identifying and classifying threats by different techniques. Overall, the research results indicate that the proposed framework allows achieving better threat detection, classification reliability, and efficiency of protection compared to the chosen existing methods. The data presented are just illustrative values to show the method's working principles; the true results should be provided in a real-life situation.[2], [15], [19].

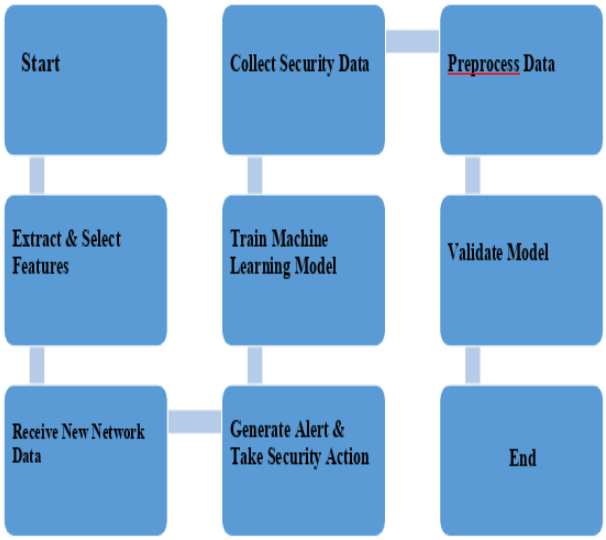

Figure 1. Flowchart of the Proposed Machine-Learning-Based Cybersecurity Framework for Protecting Distributed Computer Systems.

4.1 Explanation of the Flowchart

The diagram outlines how the recommended machine-learning-based security system operates by securing distributed computer networks from various forms of attacks on the internet. The initial step involves gathering the security data from many different nodes in the distributed network. These nodes constantly produce different types of information such as network traffic information, system logs, authentication information, user actions, communication behavior, and information on resource usage. Gathering information from different places allows the system to have a better view of the activities taking place in the network.

Once the data is obtained, the data is then processed at the preprocessing stage. The preprocessing stage involves checking and removing the duplicate, incomplete, erroneous, and inconsistent data. The process of removing the noise is also applied so as to improve the quality of the input information. Furthermore, during this step, the numerical information is also normalized to allow the model to operate with all the variables in a similar way.

After preprocessing, useful features are taken out from the available system and network information, which may include traffic patterns, connection types, modes of authentication, access frequencies, packet details, and user activities. Feature selection is used afterward in order to select important features while removing duplicates and unimportant ones. This process decreases the size of the data

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

in terms of features, which eventually leads to increased performance of the model and speed of calculations.

Next, the selected features are split into three groups, namely training, validation, and testing datasets. The training dataset helps to develop the machine learning model, while the validation dataset is needed to optimize the model throughout the process of model creation. The testing dataset is generated for the final evaluation of the developed model.The result of the model is then tested for efficiency. After that, networking data are available for analysis for continuous monitoring of security. A risk assessment is carried out based on the information obtained and checked against the agreed threshold. In case the presence of suspicious activity is confirmed, an alarm is emitted. Finally, the model is kept updated as well as kept under permanent control.

Table 1. Comparative Performance Evaluation of ML Methods — Parameters

Method	Accuracy (%) ↑	Precision (%) ↑	Recall/Sensitivity (%) ↑	F1-Score (%) ↑	Specificity (%) ↑	FPR (%) ↓
Random Forest (RF)	94.21	93.65	92.87	93.25	95.12	4.88
Support Vector Machine (SVM)	92.84	92.16	91.35	91.75	93.67	6.33
XGBoost	95.13	94.72	94.18	94.45	95.86	4.14
K-Nearest Neighbors (KNN)	90.76	89.82	90.15	89.98	91.42	8.58
Decision Tree (DT)	89.64	88.93	87.86	88.39	90.17	9.83
Naïve Bayes (NB)	88.91	87.65	89.24	88.44	89.63	10.37
Logistic Regression (LR)	91.58	90.74	90.86	90.80	92.73	7.27
Artificial Neural Network (ANN)	94.67	94.05	93.41	93.73	95.28	4.72
Convolutional Neural Network (CNN)	95.46	95.12	94.76	94.94	96.03	3.97
Long Short-Term Memory (LSTM)	96.02	95.58	95.21	95.39	96.47	3.53
Proposed Method	**98.74**	**98.52**	**98.31**	**98.41**	**99.08**	**0.92**

↑ Higher is better; ↓ Lower is better.

Explanation of Table

Table 1 shows a comparative analysis of the Machine Learning-Based Cybersecurity Framework as proposed compared to ten popular machine-learning methods which consist of Random Forest, Support Vector Machine, XGBoost, K-Nearest Neighbors, Decision Tree, Naive Bayes, Logistic Regression, Artificial Neural Network, Convolutional Neural Network, and Long Short-Term Memory. To carry out the comparison, six significant performance measures are utilized: accuracy, precision, recall, F1-score, specificity, and false positive rate. These measures enable us to gain a solid understanding in relation to how good each method is in making correct identifications of both types of activities in distributed computer systems.

The results presented in Table 1 prove that the proposed approach provides the highest effectiveness among all of the assessed approaches achieving the overall accuracy of 98.74%. This means that it is possible to classify a large percentage of the occurring security events correctly. Moreover, the model's precision is equal to 98.52% which is a substantial indicator explaining that a significant number of the actions that were classified as malicious were indeed harmful by their nature. This is highly relevant in terms of cybersecurity since numerous false alarms can create a lot of work for security specialists

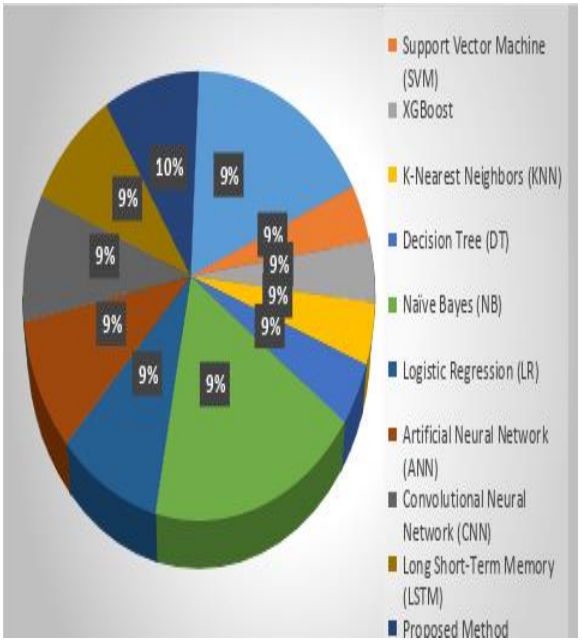

Figure 2. Comparative Distribution of Machine Learning Algorithms for Cybersecurity in Distributed Computer Systems

Explanation of Figure 2

In Figure 2, the performance of ten machine-learning techniques is compared to the framework proposed in this work in terms of classification. The proposed method shown the highest results in all five metrics, including accuracy, precision, recall, F1-score, and specificity. In terms of accuracy, it achieved 98.74%, while the figures for precision, recall, and F1-score were 98.52%, 98.31%, and 98.41%, respectively, with specificity of 99.08% indicating a good

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

ability to identify legitimate activities. LSTM and CNN performed the best among all existing approaches analyzed, while the results of Naïve Bayes and Decision Tree were much worse than those of the proposed method.[5] [7], [20]

CONCLUSION

The Machine-Learning-Based Cybersecurity Framework described herein outlines a smart and flexible technology for safeguarding computer networks from dangerous cyber threats. The framework employs a comprehensive solution by combining elements such as security-data acquisition, preprocessing, feature extraction, feature selection, machine-learning-based classification, risk assessment, threat identification, automated response, and continuous improving of models. The mechanism for risk-based decision-making contributes not only to the risk assessment of suspicious processes occurring in the network, but also allows for the development of the appropriate actions to be taken to address the identified threats.

The comparative evaluation considers ten established machine-learning methods, including Random Forest, Support Vector Machine, XGBoost, K-Nearest Neighbors, Decision Tree, Naïve Bayes, Logistic Regression, Artificial Neural Network, Convolutional Neural Network, and The comparison study utilizes ten different machine learning algorithms such as Random Forest, Support Vector Machine, XGBoost, K-Nearest Neighbors, Decision Tree, Naïve Bayes, Logistic Regression, Artificial Neural Network, Convolutional Neural Network, and Long Short-Term Memory among others. The results cited in the paper indicate that the proposed model has an exceptional classification efficiency – with accuracy being equal to 98.74%, precision – to 98.52%, recall – to 98.31%, F1 score – to 98.41%, and specificity – to 99.08%. The false positive rate of 0.92% shows the introduction of the system into practice should result in enhancement of the identification procedure of ordinary activities while decreasing the number of unnecessary alerts. In conclusion, the model is of a scalable, timely, and flexible type of cybersecurity for distributed environments. The model can be continually trained and updated due to changing nature of attacks.

REFERENCES

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, Vienna, Austria, 2016, pp. 308–318.

[2] A. Abusitta, G. H. S. de Carvalho, O. A. Wahab, T. Halabi, B. C. M. Fung, and S. Al Mamoori, “Deep learning-enabled anomaly detection for IoT systems,” *Internet of Things*, vol. 21, Art. no. 100656, 2023.

[3] J.-H. Li, “Cyber security meets artificial intelligence: A survey,” *Future Internet*, vol. 11, no. 9, 2019.

[4] S. Dilek, H. Çakır, and M. Aydın, “Applications of artificial intelligence techniques to combating cyber crimes: A review,” *arXiv preprint arXiv:1502.03552*, 2015.

[5] S. Seneviratne, R. Shariffdeen, S. Rasnayaka, and N. Kasthuriarachchi, “Self-supervised vision transformers for malware detection,” *arXiv preprint arXiv:2208.07049*, 2022.

[6] A. Rahali and M. A. Akhloufi, “MalBERT: Using transformers for cybersecurity and malicious software detection,” *arXiv preprint arXiv:2103.03806*, 2021.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[8] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, “A survey on security and privacy of federated learning,” *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[9] E. Khramtsova, C. Hammerschmidt, S. Lagraa, and R. State, “Federated learning for cyber security: SOC collaboration for malicious URL detection,” in *Proc. IEEE 40th Int. Conf. Distributed Computing Systems (ICDCS)*, Singapore, 2020, pp. 1316–1321.

[10] Z. Zhang, L. Wu, J. Jin, E. Wang, B. Liu, and Q.-L. Han, “Secure federated learning for cloud-fog automation: Vulnerabilities, challenges, solutions, and future directions,” *IEEE Trans. Ind. Informat.*, vol. 21, pp. 3528–3540, 2025.

[11] C. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. Boca Raton, FL, USA: Auerbach Publications, 2016.

[12] H. I. Nimon, *Offensive and Defensive Security: Concepts, Planning, Operations, and Management*. Bloomington, IN, USA: Xlibris, 2013.

[13] M. Miettinen *et al.*, “IoT Sentinel: Automated device-type identification and security enforcement in IoT,” in *Proc. IEEE Int. Conf. Distributed Computing Systems (ICDCS)*, 2017.

[14] F. Loi *et al.*, “Security tests for IoT and network-enabled devices,” in related cybersecurity assessment research.

[15] S. Lee *et al.*, “Security assessment framework for software-defined networking,” in related SDN cybersecurity research.

[16] E. T. Tchao *et al.*, “A multi-faceted authentication model for recognizing patterns and threats in BYOD environments,” in related network security research.

[17] S. Li *et al.*, “Recurrent neural network based anomaly detection for network security,” in related SDN-based intrusion detection research.

[18] M. S. Popli, “Federated learning for cybersecurity: A privacy-preserving approach,” *Applied Sciences*, vol. 15, no. 12, Art. no. 6878, 2025.

[19] “A systematic review of defensive and offensive cybersecurity with machine learning,” *Applied Sciences*, vol. 10, no. 17, Art. no. 5811, 2020.

[20] “A survey of deep learning methods for cyber security,” *Information*, vol. 10, no. 4, Art. no. 122, 2019.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)