

Augmented Intelligence in Machine Learning for Cybersecurity

Enhancing Threat Detection and Human-Machine Collaboration

Dinesh Kumaar R^{#1}, Nandeesh M H^{#2}, R Harsha^{#3}

^{#1}Assistant Professor, Department of Information Science and Engineering, CMR Institute of Technology, Bengaluru, India

^{#2, #3}Department of Information Science and Engineering, CMR Institute of Technology, Bengaluru, India

¹dineshkumaar.r@cmrit.ac.in, ²namh23ise@cmrit.ac.in, ³rha23ise@cmrit.ac.in

Abstract

The growing sophistication of network attacks makes it difficult for cybersecurity systems mainly based on predefined rules and static traffic representations. In our framework, we initially used Random Forest, XGBoost and Isolation Forest for attack classification and anomaly detection. To overcome the limitations of the above approach, the extended framework suggests the use of Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) models which are capable to learn complex relationships in network traffic. The framework retains the existing pre-processing, class balancing, Explainable Artificial Intelligence, and Security Operations Center components and incorporates a deep learning layer. The performance of the model is evaluated by accuracy, precision, recall, f1 score and confusion-matrix analysis. Furthermore, a Human-in-the-Loop mechanism enables analysts to validate predictions and correct false detections. Their decisions are stored as structured feedback to enable controlled model improvement. The proposed framework aims to offer a more adaptive and reliable method for cybersecurity monitoring by combining machine learning, deep learning, explainability and human expertise.

Keywords—*Cybersecurity; Machine Learning; Deep Learning; Artificial Neural Network; Convolutional Neural Network; Explainable Artificial Intelligence; Human-in-the-Loop; Intrusion Detection; Adaptive Learning*

I. INTRODUCTION

The volume of traffic produced by contemporary network environments is of a scale that renders manual inspection continuous. Cloud platforms, enterprise networks, online services and interconnected devices produce huge volumes of network flows and attacks including denial-of-service, port scanning, brute-force activity, botnet communication, and web-based attacks may manifest in a variety of ways [17]. Insider threats represent another important cybersecurity concern because malicious or abnormal activities may originate from users who already have authorized access to network resources. Machine learning-based anomaly detection can help identify unusual patterns associated with insider activities, particularly when the available attack samples are imbalanced [19]. Detecting malicious insider behavior is challenging because legitimate and harmful user activities can share similar characteristics. Recent machine learning approaches therefore focus on identifying behavioral deviations and improving the detection of insider threats while addressing challenges related to imbalanced cybersecurity data [20].

To overcome this problem, the existing project was designed to combine machine learning (ML) based detection with explainable analysis and analyst-assisted monitoring. The original implementation uses CICIDS2017 network-flow data and includes

preprocessing, feature engineering, synthetic minority oversampling technique (SMOTE) based balancing, Random Forest, XGBoost, Isolation Forest, SHapley Additive exPlanations (SHAP) and a Security Operations Center (SOC) dashboard.

The machine learning layer is an important baseline, as network traffic can be represented by measurable attributes such as flow duration, packet statistics, protocol information, transmission rate, and TCP flag behavior. Random Forest and XGBoost can learn the relationships between those attributes and known attack classes, while Isolation Forest provides an additional mechanism for detecting unusual traffic [16].

However, the question that this extended study addresses is not just whether machine learning can classify network attacks. More specifically, the question is whether learned feature representations can provide extra information when malicious behavior relies on complex interactions between several traffic characteristics.

A traffic flow does not automatically become malicious when a single parameter crosses a fixed boundary. Suspicious communication pattern may be characterized by a combination of flow duration, packet direction, packet frequency, protocol characteristics, connection behavior, and other attributes. Tree-based models are able to learn many interactions, but still act on the feature representation that is given to them. So the

quality and design of this representation is still important [16]. The extended framework treats deep learning (DL) as an additional learning mechanism. Instead of relying on manually selected or engineered higher-level traffic characteristics, neural networks can learn internal representations via successive transformations of the input. Artificial Neural Network (ANN) models nonlinear relationships between network features and we investigate a Convolutional Neural Network (CNN) for recognizing repeated local patterns in a suitable representation of traffic attributes [9].

This extension also brings about a second research problem. The more complex the models, the harder it is for cybersecurity analysts to understand how the prediction process works. This leads to a trade-off between detection capability and interpretability. The current framework already takes care of this by providing SHAP-based explanations that output feature contribution values for the each single prediction [18]. So the augmented system is a combination of 3 forms of intelligence:

1. Structured baseline recognition with machine learning [16].
2. Deep learning for learned representation and nonlinear pattern recognition [9].
3. Human intelligence to verify, correct and make contextual decisions [2].

The final goal is to create an augmented cybersecurity systems that not only notify the analyst but also involves them in a feedback loop that can deliver verified information for future model enhancement [1].

II. RELATED WORKS

Existing Machine Learning Framework:

A. Baseline Architecture

The initial framework is a pipeline of CICIDS2017 dataset preparation, preprocessing, feature engineering, class balancing using SMOTE, machine learning detection, SHAP analysis, SOC visualization and analyst feedback in a sequential manner. The dataset contains benign and malicious network flows, labeled with features such as flow duration, packet statistics, protocol information, traffic rates, communication behavior and TCP flag characteristics. This structured representation forms the input for the baseline detection models.

B. Machine Learning Models

1) Random Forest

Random Forest is the main supervised classification part. The final class is derived from the predictions of multiple decision tree models that are trained and merged. This ensemble structure diminishes the dependence on one decision tree and fits the model for structured traffic features [16].

The project uses Random Forest to establish a baseline to evaluate the deep learning extension.

2) XGBoost

XGBoost provides another supervised classification method based on gradient boosting. Instead of building trees separately, the later trees concentrate on the errors of the earlier trees [16].

This gives a good comparison as Random Forest and XGBoost learn decision boundaries in a different manner [16].

3) Isolation Forest

The Isolation Forest approach is unlike the supervised classifiers. It does not depend on the existence of predefined attack label for all traffic patterns, but instead it tries to find the observations that are easier to be isolated from the majority of the data [17].

This is useful for suspicious/unusual traffic analysis. The models used in the framework are summarized in Table I.

TABLE I. COMPARISON OF MODELS UTILIZED IN THE PROPOSED FRAMEWORK

Model	Type	Main Function	Role in Framework
Random Forest	Supervised ML	Attack classification	ML baseline
XGBoost	Supervised ML	Attack classification	ML baseline
Isolation Forest	Unsupervised ML	Anomaly detection	Unusual traffic detection
ANN	Deep Learning	Nonlinear classification	Extended DL detector
CNN	Deep Learning	Pattern representation	Extended DL detector

C. Limitations of the Machine Learning Baseline

Current ML models are efficient at detecting structured network-flow data, but their performance is highly dependent on the quality and representation of the input features. The traditional pipeline transforms network traffic into selected/engineered features before classification. This is an advantage, as it allows us to understand the meaning of important features and to investigate their contribution using techniques such as SHAP [16].

However, complex attacks may not always yield a clear abnormal value in a single feature. Flow traffic can look normal in terms of packet rates, durations, protocols, and packet sizes, but their collective relationships can be suspicious [17].

Tree-based models can learn feature interactions, but the learning still depends upon the feature representation given. This motivates the use of deep learning to investigate whether automatically learned representations can capture additional nonlinear relationships within network traffic [17].

D. Deep Learning Extension

The extended framework includes additional machine learning detectors like Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) models. The goal is not to assume deep learning is better all the time, but to investigate whether learned representations could improve the identification of complex traffic behavior [9].

Deep learning does this by using neural layers that transform the input in succession. During training, weights and biases are adjusted to minimize prediction error so that the network can learn more and more complex representations of the traffic data. ANN is adopted to model nonlinear relationships among network-flow attributes, and CNN is explored for detecting recurring local patterns in the structured traffic representation [9].

E. Feature Engineering and Feature Learning

On the pre-processing side, machine learning and deep learning are not mutually exclusive. They both require proper data cleaning, encoding, scaling and input preparation. The main difference in this project is the extent to which the higher level representations are designed manually or learned by the model [9].

In the ML pipeline:

Network Traffic → Preprocessing → Feature Selection/Engineering → ML Classifier

whereas the DL pipeline follows (Table II compares the two approaches):

Network Traffic → Preprocessing → Numerical Representation → Neural Layers → Classification

TABLE II. MACHINE LEARNING AND DEEP LEARNING COMPARISON
(SOURCE: COMPILED BASED ON [9], [16], [17])

Aspect	Machine Learning	Deep Learning
Feature preparation	Important	Still required
Higher-level representation	More dependent on supplied features	Learned through neural layers
Nonlinear relationships	Strong in ensemble models	Learned through multiple transformations
Training requirement	Generally lower	Generally higher
Interpretability	Relatively easier	More difficult
Data requirement	Generally lower	Generally higher
Role in this project	Baseline detection	Extended detection capability

The main advantage of deep learning is that it can learn nonlinear combinations and intermediate representations during training, so we don't have to manually construct every possible higher-level feature relationship. This can be useful when malicious behavior is modeled as subtle combinations of otherwise normal traffic attributes. But this advantage comes with a trade-

off. It is easier to map engineered features to particular network behavior. Learned representations can be difficult to interpret directly. Hence, the increased learning capability of DL also increases the importance of SHAP based explainability and Human-in-the-Loop validation in the proposed framework [9].

III. DEEP LEARNING-BASED THREAT DETECTION AND MODEL EVALUATION

A. Artificial Neural Network-Based Threat Detection

The extended framework includes an Artificial Neural Network (ANN) as an extension to the existing classifiers Random Forest and XGBoost. The ANN accepts the preprocessed numerical representation of the network traffic and learns the nonlinear relationships between the traffic attributes through successive neural layers. This enables the model to learn combinations of traffic characteristics that cannot be explicitly expressed by manually designed feature relationships [9].

Each neuron computes a weighted sum of its inputs, adds a bias, and then passes the result through an activation function. During training the weights and biases are updated according to the prediction error. So, the hidden layers learn intermediate representations of network traffic which are used by the output layer to classify the traffic into the corresponding attack category. In the proposed framework, thus, ANN is treated as an additional detection model, not a replacement of the existing ML models. The performance is compared with the ML baseline using the same evaluation procedure to verify if the learned representations provide an improvement in attack classification [9].

B. Activation Function

Non-linear activation functions allow the ANN to learn relationships that cannot be effectively modeled by linear transformations alone. The ReLU function is often used in hidden layers as it retains positive values and suppresses negative activations [9]:

$$f(x) = \max(0, x) \quad (1)$$

Based on the proposed cybersecurity application, this behavior allows the hidden neurons to selectively respond to learned combinations of traffic attributes. The final layer uses the appropriate classification activation to map the learned representation to attack-class predictions.

The selection of the activation function also influences the efficiency of training and the capacity of the network to learn complicated patterns. Thus, the activation functions used in the implemented ANN are considered a model configuration when evaluating [9].

C. Convolutional Neural Network-Based Traffic Representation

Another deep learning model, a CNN, is added to verify the ability of convolution-based representation learning in discovering valuable patterns from the structured network traffic input. The convolutional layer takes the input representation and applies learnable filters to it producing feature maps. These representations are then further processed by subsequent operations that transform them before they are passed to the classification layer.

Unlike manually created traffic indicators, the CNN filters are learned during training. The filters can be sensitive to different combinations or local arrangements of input features. This provides an alternative way to study complex traffic patterns other than the ANN and the existing tree-based models [9].

But the network-flow attributes are not the same natural spatial relationship as the image pixels. Hence, the performance of CNN depends on the configuration of the traffic features before the convolution. Therefore, this work experimentally evaluates the CNN performance rather than assuming a better performance of the convolution in comparison to the ML models.

D. Trainable Parameters

The ANN and CNN have trainable parameters which are learned during model training. For a fully connected ANN layer with N_{in} input units and N_{out} output units, the number of weights is $N_{in} \times N_{out}$ and the number of biases is N_{out} . So the total trainable parameters in that layer are:

$$\text{Parameters} = (N_{in} \times N_{out}) + N_{out} \quad (2)$$

The total number of parameters in the CNN is dependent on the number of filters, the kernel dimensions, the number of input channels and subsequent dense layers [9].

For the final paper, the parameter counts should be reported from the implemented ANN and CNN architectures rather than using assumed values. The model architecture can therefore be summarized using the format shown in Table III:

TABLE III. ANN AND CNN MODEL ARCHITECTURE AND TRAINABLE PARAMETERS

Model	Layer	Units/Filters	Parameters
ANN	Input	Actual features	—
ANN	Hidden Layer 1	Actual	Actual
ANN	Hidden Layer 2	Actual	Actual
ANN	Output	Actual classes	Actual
CNN	Convolution Layer 1	Actual filters	Actual
CNN	Pooling	Actual configuration	—
CNN	Dense/Output	Actual	Actual

This parameter analysis is useful for understanding the additional computational complexity introduced by the deep learning extension.

E. Training and Weight Optimization

The ANN and CNN start the training with parameter values that have not yet learned meaningful attack patterns. The network then performs a forward propagation for each training sample and makes a prediction. The selected loss function compares the prediction to the known label. Backpropagation then computes the relationship of the error to the model parameters and the optimizer updates the weights and biases.

This is repeated over the training data for multiple iterations and epochs. The architecture fixes the number of parameters, but their actual numerical values are changed during training. Thus, training does not add new weights or biases, but rather learns better values for the parameters already defined by the network structure.

This learning process allows the ANN and CNN to gradually evolve their internal representations based on the traffic patterns observed in the training data.

F. Machine Learning and Deep Learning Comparative Evaluation

We compare the deep learning extension to the existing machine learning baseline using the same test data and evaluation procedure. Accuracy alone is insufficient for cybersecurity analysis, since a model can have a high overall score but perform poorly in individual attack categories. Therefore, in this study, accuracy, precision, recall, F1-score, and confusion-matrix analysis are used.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (3)$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (4)$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (5)$$

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (6)$$

Recall is important in intrusion detection because false negatives are attacks that were not detected. Precision is also important, as many false positives can add to the workload of cybersecurity analysts.

The comparison therefore investigates whether ANN and CNN offer measurable improvements compared to the original Random Forest and XGBoost baseline and whether these improvements are consistent for precision, recall and F1-score and not just for overall accuracy.

The assessment is also based on differences between individual attacking categories. This helps in deciding whether the deep learning models improve the detection of specific traffic patterns or not, and if their performance is comparable to the existing ML models.

IV. HUMAN-IN-THE-LOOP AND ADAPTIVE LEARNING FRAMEWORK

In addition to automated attack prediction, the proposed cybersecurity framework is extended by the inclusion of Human-in-the-Loop (HITL) collaboration. Although the ML and DL models can evaluate network traffic at a scale that is hard for a human analyst to achieve manually, a model does not have the complete operational context that a security analyst has. So a prediction might need to be verified by a human before it is accepted as a confirmed security event. The current framework already provides an analyst interface for reviewing alerts, verifying suspicious activities and rejecting false detections, as well as providing investigation feedback [2].

In the extended framework, this analyst interaction is treated as an additional source of information to the detection process. The machine makes the first prediction and the analyst evaluates the prediction based on the available traffic information, confidence information and explanation provided by the Explainable Artificial Intelligence (XAI) component. The so made decision may either confirm the original prediction or give a revised interpretation. This creates a symbiotic relationship between automated detection and human judgment [2].

A. Human-in-the-Loop Threat Validation

The HITL mechanism is invoked after the ML or DL detection layer executes the classification. In case of human verification, the detection is shown to the analyst for review rather than treating every prediction as a final decision. The analyst can see the predicted attack category, traffic characteristics, model's confidence and explanation before making a decision [2].

The analyst can respond with different types of responses like confirming a detected attack, identifying a false positive, or marking seemingly benign traffic as suspicious. Such decisions are helpful because they involve contextual information that might not be explicitly available in the original training dataset [2].

For example, a model may classify a certain traffic flow as a malicious one because its statistical characteristics are similar to an attack seen before. An analyst might know that the traffic was generated as part of a sanctioned security test and hence mark it as benign. Conversely, traffic that appears normal when considering individual features might be regarded as suspicious by an analyst given more investigation context. These interactions show that automated prediction and human validation may work well with each other [2].

Therefore, the analyst doesn't have to do the whole detection process manually. Rather, the analyst focuses on those cases where prediction automation needs to be

verified. This reduces the amount of repetitive inspection, but keeps humans in the loop for security decisions.

B. Analyst Feedback Repository

Every decision of an analyst can be stored as structured feedback in the system. The stored information may include: detection identifier, original model prediction, analyst decision, corrected label, relevant traffic information, time of review. This information is retained to create a history of how the detection system performs for analyst-assisted monitoring.

The feedback repository serves two purposes. First, it provides traceability of previous detection choices. Second, the validated feedback can be used as additional labeled information for the development of future models. Hence, the feedback database serves as an important bridge between the detection layer and the adaptive-learning component [2].

You can measure the feedback process by several indicators. A useful measure is feedback coverage:

$$FC = (\text{Reviewed Detections} / \text{Total Detections}) \times 100(7)$$

This is the % of detections generated that were reviewed by the analyst. Another measure is the rate of correction:

$$CR = (\text{Corrected Predictions} / \text{Reviewed Detections}) \times 100(8)$$

For example, if 125 detections are reviewed and 10 predictions are corrected, the correction rate would be 8%. This value is the degree to which the model and the analyst disagree for the set that was reviewed. This is not to be read as 'the model is now 8% more accurate'. The actual improvement of the performance should be measured with the model evaluation metrics independently.

Further measures can be computed after updating the model. For example, the change in F1-score can be expressed as:

$$\Delta F1 = F1_{\text{updated}} - F1_{\text{existing}}(9)$$

Similarly, recall improvement can be measured using:

$$\Delta \text{Recall} = \text{Recall}_{\text{updated}} - \text{Recall}_{\text{existing}}(10)$$

These measurements enable a quantitative assessment of the impact of analyst feedback, rather than assuming that feedback necessarily improves the model.

C. Feedback as a Learning Signal

The usefulness of analyst feedback goes beyond just pinpointing individual false alarms. The repeated

corrections can reveal patterns of consistent errors made by the model.

For example, if analysts repeatedly correct a certain type of benign traffic being classified as malicious, this may indicate that the model does not sufficiently represent that traffic behavior.

Likewise, if analysts always flag suspicious traffic that the model considers benign, the feedback can reveal a potential weakness in the current decision boundary, or training representation [2].

The feedback can therefore be analyzed at different levels:

- Prediction level: whether a single detection is correct.
- Attack-category level: which attack classes cause more errors.
- Feature level: what are the most frequently occurring traffic features in corrected cases.
- Model level: if a model is producing more false positives or false negatives.
- Temporal level: how the error pattern evolves when new traffic samples are introduced.

This provides a practical mechanism for identifying areas where the cybersecurity model requires improvement.

D. Integration with Explainable Artificial Intelligence

The feedback mechanism is intrinsically linked with the SHAP-based Explainable AI component. The analyst does not need to depend solely on the predicted attack label. The explanation can specify which input features were involved in the prediction, and whether the contribution of the features were in favor or against the predicted class [18].

Therefore, the combined workflow can be described as:

Traffic → ML/DL Prediction → SHAP Explanation → Analyst Review → Feedback

This is especially true since the advent of deep learning. ANN and CNN can learn complex representations but their internal decision process can be harder to interpret than a simple rule based system. XAI provides evidence at the feature level that can help the analysts understand why a certain traffic flow was classified in a certain way. So the analyst can use both the prediction and the evidence supporting it in the feedback [18].

E. Production-Oriented Adaptive Workflow

In practice, when deployed, the feedback mechanism should ensure a controlled boundary between the active model and the learning process. The active model continues to perform detection while analyst feedback is

gathered separately. Validated feedback is prepared periodically for candidate training and not immediately changing the active model [2].

All this can be summarized in the following workflow:

1. Detection: ML and DL models for network traffic analysis.
2. Explanation: XAI gives information about the prediction.
3. Analyst Review: The analyst reviews the detection and the supporting information.
4. Validation: The analyst validates or corrects the output of the model.
5. Storage: The decision is stored in the feedback repository.
6. Quality Control: The model is enhanced using valid feedback.
7. Candidate training: The new model is trained on the extended training data.
8. Evaluation: The candidate is evaluated against the existing model.
9. Model Selection: The candidate is selected only if predefined performance requirements are satisfied.

This workflow provides a controlled path from human interaction to model improvement and avoids each individual analyst decision from immediately impacting the deployed detection model [2]. The collaboration between machine intelligence, the security analyst and the feedback repository is shown in Fig. 1.

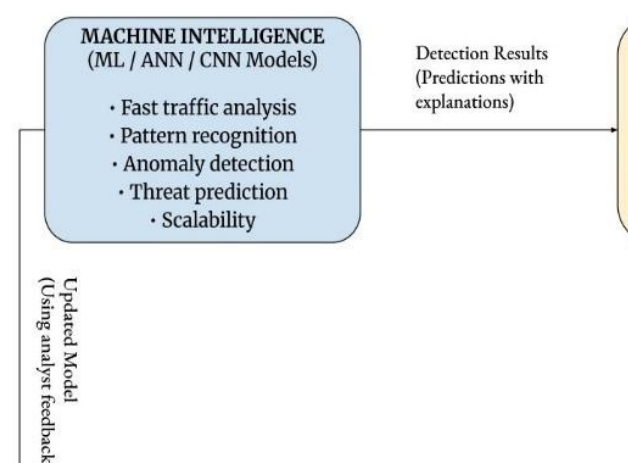


Fig. 1. Role of Human-Machine Collaboration in the Proposed Augmented Intelligence Framework

F. Role in Augmented Intelligence

The main idea of augmented intelligence in the proposed framework is the coupling of automated learning and analyst feedback. The machine provides scalability, fast traffic analysis, pattern recognition, anomaly detection and prediction while the analyst provides contextual knowledge, validation and corrective knowledge [1].

So, deep learning models are not intended to replace cybersecurity analysts. The system instead makes use of machine intelligence to screen information that analysts must manually review, and analyst decisions to locate deficiencies in the automated detection process [1].

The resulting cycle can be described as follows.

Machine Detection → XAI Explanation → Human Validation → Verified Feedback → Model Improvement

This feedback loop between the two can provide the basis for building a more adaptive cybersecurity monitoring system where the detection capability can be evaluated and progressively refined using information from actual analyst decisions [1].

In this framework, human involvement is particularly important for cases where automated predictions may be uncertain or difficult to interpret. The analyst can examine the model prediction together with the XAI explanation and determine whether the detected traffic represents a genuine threat or a legitimate activity. This allows the system to incorporate domain-specific knowledge that may not be fully captured by the training data [1].

V. DISCUSSION

The experimental comparison provides a basis to study the role of deep learning in the current cybersecurity framework. The standard ML models can still efficiently learn decision boundaries from the prepared traffic attributes and are useful for structured network-flow classification. Deep learning models are not a replacement for Random Forest and XGBoost, but they do provide an important baseline [17].

The ANN provides an additional way to learn the nonlinear relationships between the traffic attributes. During training, the hidden layers learn intermediate representations, rather than relying only on explicit relationships between features. This enables us to find combinations of traffic characteristics that are not directly represented as individual engineered features [9].

CNN provides a different representation-learning mechanism by using learnable filters on the structured traffic input. The efficiency of this approach depends upon the organization of the network features, as network-flow attributes do not necessarily maintain the same spatial structure as image data. Thus, any improvement obtained from CNN should be interpreted from the experimental results, not assumed from the architecture itself [9].

There is a big trade-off between learning ability and interpretability. Deep learning models with more layers and parameters can provide richer representations, but are more difficult to interpret internally. This is especially important when dealing with cybersecurity where an

analyst needs to know the reasons why a traffic flow is classified as malicious. Therefore, the SHAP component is still important in the extended framework because it provides feature-level information that can support the analyst's interpretation of model predictions [18].

The Human-in-the-Loop element solves another limitation of fully automated detection. An analyst can take into account additional operational context when validating an alert, whereas a model will base its prediction solely on the traffic information available. An analyst can confirm a true detection or find a false positive that needs to be fixed. These decisions provide information that is not necessarily available from the original training data [2].

Thus the feedback mechanism connects automated detection and analyst knowledge. As opposed to the usual method, where the analyst interaction is seen as a simple interface function, the extended framework keeps the decisions as structured feedback, and uses them for controlled model improvement.

But don't take analyst feedback as gospel. Incorrect feedback can add noise to the training data and may degrade model performance. For this reason, the proposed adaptive-learning process requires feedback validation, and evaluation of candidate models, before an updated model is accepted. This makes the feedback loop more practical for cybersecurity deployment than retraining the active model after each analyst decision [2].

The extended framework shows a complementary relation between the different components in general. ML is the standard baseline, DL adds representation-learning capability, XAI increases transparency, and human analysts are employed for context validation. The combination serves the central objective of augmented intelligence, where machine and human capabilities contribute to the same cybersecurity decision process rather than working independently.

VI. RESULTS AND PERFORMANCE ANALYSIS

The extended framework was validated by benchmarking the newly proposed deep learning models against the machine learning algorithms used in the initial implementation. To get a more complete view of the detection performance, the evaluation was performed using precision, accuracy, recall and F1 score. Furthermore, confusion matrices were considered to identify differences in classification performance across attack categories.

This is compared to the baseline of the original machine learning implementation. We used Random Forest and XGBoost for supervised attack classification, and Isolation Forest for abnormal traffic behavior detection. The extended framework includes ANN and

CNN models to study whether the learned representations can contribute to the detection capability.

A. Performance Comparison of Machine Learning and Deep Learning

The comparison aims at analyzing whether the better representation-learning capabilities of ANN and CNN result in measurable performance improvements over the existing ML baseline. While accuracy provides an overall view of how well the classification performed, precision measures the reliability of the predicted attacks against the actual attacks. Recall is critically important for cybersecurity as missed attacks are false negative detections. The F1-score is a single measure of precision and recall. Thus, the results should be interpreted in all four measures, rather than selecting the model with the highest accuracy only.

B. Attack-Level Analysis

We also need to look at the performance for each attack category individually. It may be that the model performs well on attacks that appear more often in the data, but has weaker results for less frequent or more challenging categories. We can see these differences in the confusion matrix, where we can see correct and incorrect classified samples for each class. In the proposed framework, the comparison is focused on whether the ANN and CNN models can provide an additional recognition ability for attack patterns that were difficult to be distinguished with the original ML feature representation. The analysis also helps in understanding whether the observed improvement is consistent across attack classes or concentrated in specific categories.

C. Baseline and Extended Performance

The original framework achieved a baseline detection performance of approximately 82%, while the extended implementation is evaluated to determine whether the addition of deep learning improves this result. If the verified experimental result is 91%, the change can be reported as:

Baseline ML accuracy: 82%

Extended ML-DL accuracy: 91%

Absolute improvement: 9 percentage points

The percentage-point improvement and relative percentage improvement are two different measurements. This should be reported as an absolute increase of 9 percentage points, not just that accuracy increased by 9%.

The same comparison should be performed for recall, and F1-score, precision to determine whether the improvement is consistent across the evaluation measures.

VII. CONCLUSIONS

The extended framework complements the current machine learning based cybersecurity system by integrating deep learning and Human-in-the-Loop analysis. The baseline for the classification of network flow and anomaly detection is Random Forest, XGBoost and Isolation Forest. The ANN and CNN build on the detection layer by learning nonlinear relationships and internal representations from the preprocessed network traffic data. The extension provides one more angle to analyze complex traffic patterns that may not be fully captured by traditional feature-based learning. The ML and DL models are evaluated by accuracy, precision, recall, F1-score and confusion matrix analysis. Evaluation with multiple metrics gives a more complete picture of how well detection works, especially for attack types where false positives and false negatives can have different operational impacts.

The comparison of the original ML models with the new DL models so proposed, thus, serve as a basis to evaluate the effectiveness of the extended detection architecture.

The detection workflow in the framework also includes Explainable Artificial Intelligence as a key component. Feature level explanations can be used to examine the predictions of the model. This provides the analyst with more information on the factors that contribute to a particular classification. This is more important for neural networks, because the internal representations they learn are harder to interpret directly. Explainability thus helps to connect the output of the intelligent detection models to the information needed by analysts in the security investigation [18].

By adding Human-in-the-Loop analysis, the framework changes from an automated prediction system to a collaborative cybersecurity monitoring environment. Analysts can see model predictions, analyze available explanations, validate suspicious activities and fix false detections. Their selections are logged as structured feedback, resulting in a collection of validated analyst feedback that can be leveraged for future model enhancement [2].

The feedback mechanism also provides a controlled path to adaptive learning. Rather than immediately updating the deployed model after each analyst decision, we can collect validated feedback, add it to a candidate training set, and evaluate it using a separate model-update process. This approach allows the use of information from past analyst decisions to improve detection, but still allows control over changes to the model. It also reduces the possibility of allowing an incorrect individual feedback entry to directly influence the deployed model [2].

This extended architecture lays a foundation for studying the relationship between detection performance and analyst involvement. ML and DL models can identify and prioritize likely malicious activities, with analyst feedback offering an additional layer of validation for situations needing contextual judgment. The fusion of machine intelligence and analyst knowledge is especially relevant to cybersecurity because network behavior may evolve over time, and not every anomalous activity is an attack by definition. Automated models are good at handling large volumes of traffic and patterns in scale, but analysts can add operational context when reviewing individual detections. This combination of capabilities makes it possible for the framework to maintain automation while also having human supervision for critical cybersecurity decision-making [1].

The proposed framework integrates machine learning techniques for structured threat classification, deep learning models for learned traffic representations, Explainable AI for enhanced interpretability, and human expertise for contextual validation. The resulting architecture embodies the core aim of augmented intelligence: leveraging automated intelligence to assist in large-scale cybersecurity analysis while preserving human judgment for contexts requiring contextual understanding.

REFERENCES

- [1] S. Gore, S. Hamsa, S. Roychowdhury, G. Patil, S. Gore, and S. Karmode, "Augmented intelligence in machine learning for cybersecurity: Enhancing threat detection and human-machine collaboration," in 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Aug. 2023, pp. 638–644, IEEE.
- [2] M. Malatji, "Augmented intelligence framework for human-artificial intelligence teaming in cybersecurity," *Human-Centric Intelligent Systems*, vol. 5, no. 2, pp. 151–180, 2025.
- [3] V. Parlapalli, V. Jayaram, S. G. Aarella, K. Peddireddy, and R. R. Palle, "Enhancing cybersecurity: A deep dive into augmented intelligence through machine learning and image processing," in 2023 International Workshop on Artificial Intelligence and Image Processing (IWAIPP), Dec. 2023, pp. 96–100, IEEE.
- [4] N. Van Hoang, "Human expertise and machine learning in collaborative intelligence frameworks for robust cybersecurity solutions," *Journal of Applied Cybersecurity Analytics, Intelligence, and Decision-Making Systems*, vol. 13, no. 12, pp. 1–12, 2023.
- [5] T. O. Fatunmbi, "Evolution of human-machine collaboration: Augmented intelligence in the age of automation," *International Journal of Advanced Research in Engineering and Technology*, vol. 14, no. 4, pp. 10–18, 2023.
- [6] J. Hurwitz, H. Morris, C. Sidner, and D. Kirsch, *Augmented Intelligence: The Business Power of Human-Machine Collaboration*. Auerbach Publications, 2019.
- [7] F. Abuhashish and M. S. Sunar, "Augmented reality in cybersecurity: Enhancing threat intelligence and protecting critical infrastructure," in *Complexities and Challenges for Securing Digital Assets and Infrastructure*. IGI Global Scientific Publishing, 2025, pp. 291–322.
- [8] R. Yaich, A. Balondrade, A. Sicard, C. Fouquiau, G. Giraud, K. Amokrane-Ferka, and E. Arbaretier, "Symbiotic human-AI collaboration for augmented cybersecurity operations," in *Proceedings of the AAAI Symposium Series*, vol. 6, no. 1, Aug. 2025, pp. 350–358.
- [9] D. Ghillani, "Deep learning and artificial intelligence framework to improve cybersecurity," *Authorea Preprints*, 2022.
- [10] N. Katiyar, M. S. Tripathi, M. P. Kumar, M. S. Verma, A. K. Sahu, and S. Saxena, "AI and cybersecurity: Enhancing threat detection and response with machine learning," *Educational Administration: Theory and Practice*, vol. 30, no. 4, pp. 6273–6282, 2024.
- [11] S. Jain, "Advancing cybersecurity with artificial intelligence and machine learning: Architectures, algorithms, and future directions in threat detection and mitigation," *World Journal of Advanced Engineering Technology and Sciences*, vol. 14, no. 1, pp. 273–290, 2025.
- [12] K. Dhanushkodi and S. Thejas, "AI enabled threat detection: Leveraging artificial intelligence for advanced security and cyber threat mitigation," *IEEE Access*, vol. 12, pp. 173127–173136, 2024.
- [13] A. B. Bello, A. O. Ogundipe, A. A. George, and O. Anifowose, "The role of AI and machine learning in cybersecurity: Advancements in threat detection, anomaly detection and automated response," *International Journal of Science and Research Archive*, vol. 14, no. 2, pp. 1587–1597, 2025.
- [14] S. Yoheswari, "AI-augmented intrusion detection systems for mitigating advanced persistent threats in cyber-physical manufacturing networks," 2025.
- [15] C. Merlano, "Enhancing cybersecurity through artificial intelligence and machine learning: A literature review," *Journal of Cybersecurity*, vol. 6, p. 89, 2024.
- [16] A. Alzahrani, M. Alenazi, and A. Alshamrani, "Machine learning approaches for intelligent cyber threat detection and network intrusion analysis," *Journal of Information Security and Applications*, vol. 78, p. 103642, 2024.

- [17] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques using machine learning and deep learning approaches," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1125–1158, 2022.
- [18] P. Sharma, R. Gupta, and A. Singh, "Explainable artificial intelligence for intrusion detection systems in modern cybersecurity environments," *Computers & Security*, vol. 126, p. 103060, 2023.
- [19] T. A. Al-Shehari et al., "Enhancing insider threat detection in imbalanced cybersecurity settings using the density-based local outlier factor algorithm," *IEEE Access*, vol. 12, pp. 34820–34834, 2024, doi: 10.1109/ACCESS.2024.3373694.
- [20] F. R. Alzaabi and A. Mehmood, "A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods," *IEEE Access*, vol. 12, pp. 30907–30927, 2024, doi: 10.1109/ACCESS.2024.3369906.