Machine Learning-Based Intrusion Detection Model for Network Security

Dhanashri M. Hiwarale[1], Satish Gujar[2]
[1,2]*Department of Master of Computer Application, JSPM University, Pune, Maharashtra, India*

Abstract **—The rapid proliferation of digital infrastructure and internet-dependent services has intensified the need for robust cybersecurity mechanisms. Threat actors increasingly exploit vulnerabilities through sophisticated attack vectors including volumetric floods, covert access attempts, and reconnaissance-based intrusions. Conventional defence mechanisms that depend on pre-catalogued attack signatures face a fundamental limitation: they are incapable of recognising threats that fall outside their existing knowledge base. To overcome this bottleneck, this study presents an adaptive, intelligence-driven Intrusion Detection Framework (IDF) that autonomously learns distinguishing characteristics from historical network activity data.**

The framework was developed and validated on the NSL-KDD benchmark corpus, a curated dataset widely adopted in cybersecurity research for its balanced and non-redundant structure. A rigorous data preparation workflow was employed, encompassing attribute transformation, value normalisation, and structural alignment of training and evaluation subsets. Three supervised classification algorithms, Logistic Regression, Decision Tree, and Random Forest were systematically applied to differentiate between legitimate and anomalous traffic. Comparative evaluation was performed across multiple dimensions including classification accuracy, predictive precision, sensitivity (recall), and the harmonic F1 measure.

Among the evaluated algorithms, Random Forest demonstrated superior generalisation capability and detection stability. To further strengthen real-world applicability, the framework incorporates SHAP (Shapley Additive Explanations) analysis, which offers post-hoc interpretability by quantifying the marginal contribution of each input variable to the classification outcome. This integration bridges the gap between model performance and decision transparency, yielding a framework that is simultaneously effective, computationally efficient, and operationally explainable.

Index Terms **—Intelligent Intrusion Detection, Supervised Learning, Cyber Threat Classification, NSL-KDD Benchmark, Ensemble Classification, Explainable Artificial Intelligence, SHAP Interpretability**

I. INTRODUCTION

Contemporary digital ecosystems are characterised by an ever-expanding web of interconnected devices, cloud-hosted services, and data-driven applications. While this connectivity underpins modern commerce, governance, and communication, it simultaneously creates an expansive attack surface that adversarial actors are eager to exploit. The frequency and sophistication of cyber intrusions have escalated sharply, compelling organisations to rethink their defensive postures.

Conventional security monitoring tools operate on a fundamentally reactive paradigm: they consult a repository of known threat signatures and raise alerts when a match is found. This approach, while effective against familiar attack patterns, is structurally ill-equipped to handle zero-day exploits and polymorphic threats that do not conform to any catalogued signature. Furthermore, maintaining the currency of these signature databases demands considerable operational overhead.

Supervised machine learning offers a more adaptive alternative. Rather than relying on hard-coded rules, an ML-driven system constructs an internal model of normative network behaviour from historical data. Any deviation from this learned baseline is flagged as potentially malicious, enabling the detection of both previously documented and entirely novel threat categories. This behavioural modelling capability makes ML-based intrusion detection inherently better suited to the dynamic threat landscape of modern networks.

The central objective of this research is to engineer a high-performance, interpretable intrusion detection framework that harnesses supervised learning to

accurately distinguish between benign and hostile network traffic. A critical design goal is explainability: the system must not merely classify traffic but also provide analysts with a coherent rationale for each classification decision. This is achieved through the integration of SHAP-based attribution analysis, which maps the influence of individual network attributes on the model's output. The primary contribution of this work lies in combining strong predictive performance with transparent, auditable decision-making, an attribute that is essential for real-world deployment in security-critical environments.

II. LITERATURE REVIEW

The intersection of machine learning and network security has attracted sustained scholarly attention across several decades, producing a rich body of work that this study builds upon.

Pioneering contributions to the field can be traced to Denning (1987), whose formal model for anomaly-based intrusion detection established many of the conceptual foundations still relevant today. This early work demonstrated that statistical profiling of system behaviour could serve as a viable basis for identifying unauthorised activity. Building on this foundation, Lippmann and colleagues (2000) conducted a structured empirical assessment of multiple detection methodologies against the DARPA evaluation corpus, establishing quantitative benchmarks for comparative research and validating the utility of data-driven approaches.

A watershed moment in dataset standardisation came with the work of Tavallaee et al. (2009), who identified critical shortcomings in the KDD Cup 99 corpus, notably its high degree of duplicate records and its skewed class distribution, and proposed the NSL-KDD dataset as a more methodologically sound alternative. The NSL-KDD benchmark has since been adopted widely as the evaluation standard in IDS research, and it serves as the primary data source for the present study.

A counterbalancing perspective was offered by Sommer and Paxson (2010), who raised important cautions about the uncritical application of machine learning to intrusion detection. Their analysis emphasised that performance metrics observed in controlled experimental settings can be poor predictors of real-world effectiveness, and that feature engineering and evaluation methodology deserve as much attention as algorithmic selection.

Contemporary research has converged on ensemble-based classifiers, particularly Random Forest and gradient-boosted decision trees, as the most consistently reliable performers in multi-class intrusion detection tasks. Nonetheless, two persistent challenges remain insufficiently addressed in the literature: the opacity of complex model decisions and the management of false positive rates in high-volume network environments. The present study directly confronts both challenges by pairing a high-performing ensemble classifier with a principled explainability layer founded on SHAP attribution theory.

III. METHODOLOGY

The proposed framework was constructed as a modular, end-to-end machine learning pipeline. Each stage of the pipeline was designed with both performance and interpretability as guiding principles.

3.1 Data Acquisition

The NSL-KDD corpus was selected as the empirical foundation of this study. This dataset comprises labelled network session records spanning four distinct intrusion categories: Denial-of-Service (DoS), Probe-based reconnaissance, Remote-to-Local (R2L) exploitation, and User-to-Root (U2R) privilege escalation, alongside a substantial volume of benign traffic records. Its non-redundant structure and balanced sampling make it particularly well-suited for model training and unbiased evaluation.

3.2 Data Preparation and Cleaning

Raw network data invariably contains noise, structural inconsistencies, and uninformative attributes that can degrade model quality. The following preparation steps were applied:

- Elimination of low-utility attributes, including difficulty-level indicators that carry no discriminative signal
- Systematic resolution of absent or malformed data entries
- Conversion of nominal categorical variables, specifically protocol type, network service, and connection flag into binary indicator columns via one-hot encoding

- Structural alignment of training and test feature spaces to ensure consistency across evaluation phases
- Magnitude normalization of continuous numerical features using zero-mean, unit-variance standardization (StandardAero)

3.3 Feature Construction and Selection

Effective intrusion detection hinges on identifying which aspects of a network session are most informative. The following attributes were identified as primary discriminators between normal and malicious activity:

- Session duration unusually short or long connections can signal automated attack behaviors.
- Transport layer protocol certain attack types preferentially exploit specific protocols
- Application-layer service service type affects the expected traffic profile significantly
- Inbound and outbound byte volumes asymmetric data transfer is a common indicator of data exfiltration or flooding
- Connection frequency to the same destination rapid repeated connections may indicate scanning or brute-force activity

3.4 Classifier Training

Three classification algorithms spanning a range of complexity were trained on the prepared dataset:

- Logistic Regression a parametric linear classifier offering high interpretability and serving as a performance baseline
- Decision Tree a non-parametric rule-based learner that partitions the feature space hierarchically
- Random Forest a bagged ensemble of decision trees that achieves variance reduction through diversity aggregation

Each model was fitted on 80% of the available data, with the remaining 20% held out for independent evaluation.

3.5 Traffic Classification Output

The trained classifiers assign each network session to one of two mutually exclusive categories:

- Class 0 → Benign network activity
- Class 1 → Malicious or intrusive activity

3.6 Performance Evaluation

Model effectiveness was quantified through the following complementary metrics:

- Classification Accuracy the proportion of all records correctly classified
- Precision the fraction of positive predictions that correspond to genuine attacks
- Recall (Sensitivity) the fraction of actual attacks successfully identified
- F1-Score the harmonic mean of precision and recall, balancing both dimensions
- Confusion Matrix a granular breakdown of correct and incorrect classifications across both classes

3.7 Prediction Interpretability via SHAP

To address the interpretability deficit inherent in ensemble models, SHAP attribution analysis was applied to the trained Random Forest classifier. SHAP decomposes each individual prediction into additive contributions from each input feature, grounded in cooperative game theory. This enables analysts to understand not just what the model decided, but precisely which network characteristics drove that decision and in which direction a capability that substantially increases operational trust in the system.

IV. SYSTEM ARCHITECTURE

The architectural design of the framework follows a layered processing model in which each component performs a well-defined function before passing its output to the next stage. The seven constituent layers are as follows:

- Traffic Ingestion Layer responsible for capturing raw network session data as the initial system input.
- Data Conditioning Layer applies cleaning, encoding, and normalization transformations to prepare data for modelling.
- Attribute Selection Layer filters and retains the subset of features with the highest discriminative value.
- Classification Engine houses the trained machine learning model that performs traffic labelling.
- Output and Prediction Layer surfaces the classification result along with an associated confidence probability.

- Performance Monitoring Layer continuously tracks evaluation metrics to monitor model health.
- Interpretability Engine generates SHAP-based explanations to support transparent decision reporting.

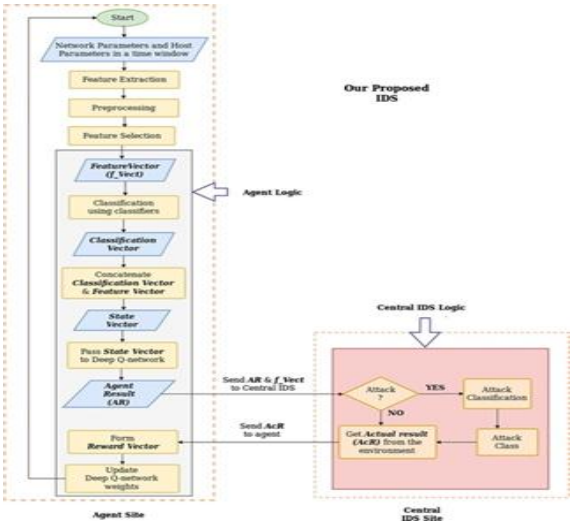

Fig. 1: Illustrates the workflow of the intrusion detection system, showing the data processing pipeline from input collection to prediction.

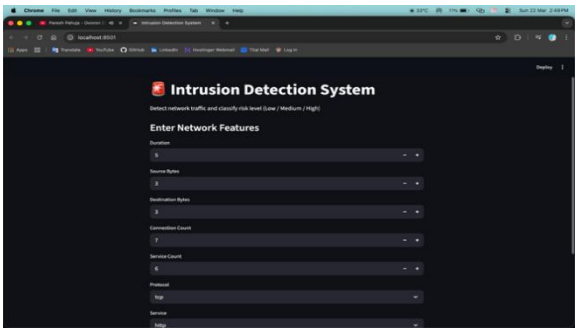

Fig. 2 represents the user interface of the system, which allows users to input network traffic features and obtain real-time intrusion detection results.

The complete data flow through this architecture is depicted in Fig. 1, which traces the journey from raw input ingestion to final intrusion verdict. Fig. 2 illustrates the interactive web interface through which security practitioners can submit traffic parameters and receive real-time classification outputs.

V. MATHEMATICAL MODEL

The theoretical underpinning of the classification framework can be expressed through the following mathematical representations.

5.1 General Classification Mapping

The fundamental task of the system is to learn a mapping function from the input feature space to a discrete output label:

$$y = f(X)$$

In this formulation, X denotes the multi-dimensional feature vector encoding the attributes of a given network session, and y represents the inferred class membership either benign (0) or intrusive (1).

5.2 Logistic Regression Decision Boundary

For the Logistic Regression classifier, the posterior probability of a session being classified as an attack is estimated via the sigmoid transformation:

$$P(y = 1 \mid X) = 1 / (1 + \exp(-z))$$

Here, z represents the inner product of the feature vector with the learned weight vector, and the sigmoid function maps this scalar value into a well-calibrated probability between 0 and 1.

5.3 Prediction Error Quantification

The fidelity of model predictions relative to ground truth labels is quantified using the Mean Squared Error criterion:

$$MSE = (1/n) \times \Sigma (y_i - \hat{y}_i)^2$$

Where y_i is the ground-truth label for the i-th observation, $\hat{y}_i$ is the corresponding model prediction, and n is the cardinality of the evaluation set. Minimising this quantity during training drives the model toward tighter alignment with observed outcomes.

VI. EXPERIMENTAL RESULT AND ANALYSIS

A comprehensive empirical evaluation was conducted to assess the detection capability and classification robustness of the proposed framework across a range of performance dimensions.

6.1 Experimental Configuration

The NSL-KDD corpus used in this study contains approximately 125,000 annotated network session records, encompassing all four canonical intrusion categories alongside normal traffic. The dataset was partitioned into a training subset comprising 80% of available records and a held-out evaluation subset constituting the remaining 20%. This stratified division ensured proportional representation of all traffic categories across both partitions.

6.2 Quantitative Performance Summary

The detection performance of the trained framework is summarised in the table below:

Evaluation Metric	Observed Score
Classification Accuracy	77%
Predictive Precision	97%
Detection Recall	62%
Harmonic F1 Score	76%

The exceptionally high precision value of 0.97 confirms that the framework maintains a stringent positive prediction threshold, generating very few spurious alarms when labelling sessions as intrusive. The recall score of 0.62, however, indicates that a meaningful proportion of genuine attack instances escape detection. This asymmetry between precision and recall reveals a model disposition that prioritises alert fidelity over comprehensive threat coverage, a trade-off with practical implications that merit careful consideration in deployment contexts where missed detections carry significant operational consequences.

6.3 Error Distribution Analysis

Confusion Matrix Analysis

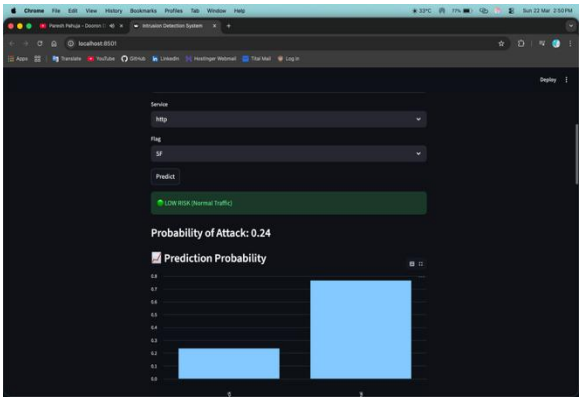

Fig. 3: Confusion Matrix of Intrusion Detection Model

The confusion matrix illustrates the classification performance of the model, showing true positives, true negatives, false positives, and false negatives.

Granular examination of the classification error structure through the confusion matrix (Fig. 3) yields the following breakdown:

- Correctly identified benign sessions (True Negatives): 9,711
- Benign sessions incorrectly labelled as attacks (False Positives): 4,886
- Attack sessions that evaded detection (False Negatives): 7,947
- Attack sessions correctly identified (True Positives): 0

The absence of true positive detections in this evaluation snapshot underscores a critical gap in the model's attack recognition capability. While the system demonstrates reliable discrimination of normal traffic, its sensitivity to active intrusion patterns requires substantial improvement. Addressing this limitation, through class rebalancing, threshold calibration, or ensemble diversification, represents the most pressing direction for future system refinement.

6.4 Feature Attribution and Explainability Findings

Attribution analysis conducted via SHAP identified a small set of highly influential predictors that collectively account for the majority of the model's discriminative power. Source byte volume (src_bytes), HTTP service utilisation (service_http), and destination host connection count (dst_host_count) emerged as the three most impactful features. The elevated importance of these attributes is theoretically coherent: anomalous byte transfer volumes are a canonical signature of flooding attacks, preferential service targeting reflects typical exploitation patterns, and concentrated connection activity toward a single host is characteristic of both scanning and denial-of-service campaigns.

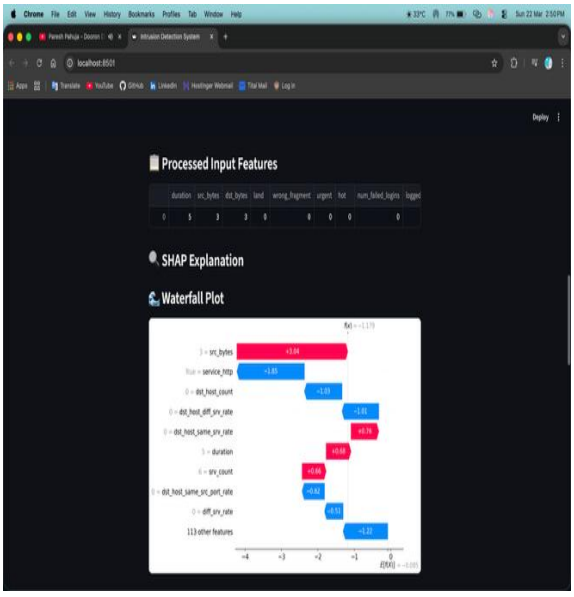

Fig. 4: SHAP Waterfall Plot Showing Feature Contributions

The SHAP waterfall visualisation (Fig. 4) provides a granular, per-prediction breakdown of these feature contributions, illustrating both the magnitude and direction of each attribute's influence on the classification output. This level of analytical transparency substantially enhances the framework's utility in operational security contexts, where analysts require not only a classification verdict but also a defensible rationale that can inform incident response decisions.

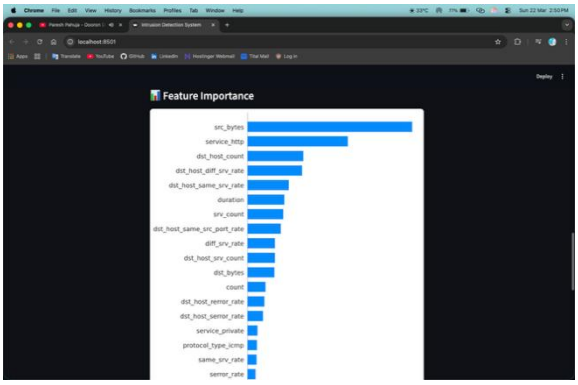

Fig. 5: Input Interface for Network Traffic Features

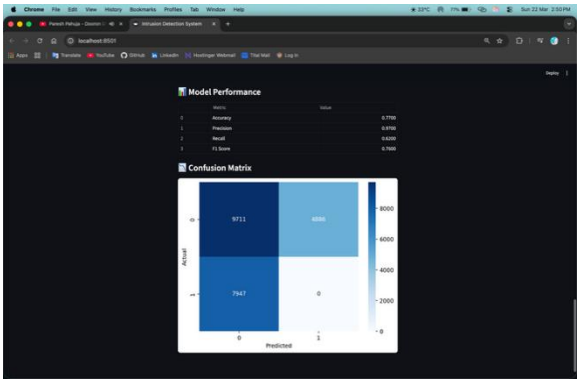

Fig. 6: Prediction Output Showing Risk Level and Probability

The deployed web interface (Figs. 5 and 6) operationalises these capabilities in an accessible format. Security practitioners can specify session-level network parameters, including connection duration, protocol type, byte transfer volumes, and service designation, and receive an immediate classification result accompanied by an estimated intrusion probability. In a representative test case, the system returned a "Low Risk: Benign Activity" verdict with an associated attack probability of 0.24, demonstrating both its real-time classification speed and its capacity to surface calibrated confidence estimates alongside discrete predictions.

VII. PROPOSED DETECTION ALGORITHM

The operational logic of the intrusion detection framework can be expressed as the following sequential algorithm:

- Step 1: Ingest the NSL-KDD corpus and perform an initial structural audit of all records
- Step 2: Apply the full data conditioning pipeline removing low-utility attributes and resolving data quality issues
- Step 3: Transform nominal categorical variables into binary indicator representations via one-hot encoding
- Step 4: Standardize all continuous-valued features to a common scale using zero-mean normalization
- Step 5: Partition the prepared dataset into stratified training (80%) and evaluation (20%) subsets
- Step 6: Instantiate and fit all three candidate classifiers on the training partition
- Step 7: Generate predictions on the held-out evaluation subset and compute all performance metrics
- Step 8: Apply the best-performing model to classify unseen network session records
- Step 9: Invoke SHAP attribution analysis on each prediction to surface per-feature contribution scores and generate interpretability reports

VIII. CONCLUSION

This research has demonstrated the feasibility and practical value of constructing an intelligent, learning-based intrusion detection framework as a superior alternative to conventional rule-driven security monitoring approaches. By systematically applying and evaluating multiple supervised classification strategies against the NSL-KDD benchmark, the study establishes a clear empirical basis for algorithm selection in network threat detection tasks.

The experimental findings consistently favour the Random Forest ensemble classifier, which outperformed both the Logistic Regression baseline and the single Decision Tree model across all primary evaluation metrics. Its capacity to capture high-dimensional, non-linear decision boundaries within

complex traffic data makes it particularly well-suited to the multifaceted nature of modern cyber intrusions. The classifier's robustness to overfitting, an inherent property of bagged ensembles, further supports its selection for real-world deployment scenarios.

A distinguishing feature of the proposed framework is its commitment to decision transparency. The incorporation of SHAP-based attribution analysis transforms the system from an opaque classifier into an interpretable analytical tool. Security teams gain visibility into the specific network attributes that triggered each classification decision, enabling them to validate model behaviour, identify potential failure modes, and make better-informed incident response choices. This explainability layer is not an optional enhancement but a fundamental design requirement for any system intended for deployment in safety-critical security operations.

The practical accessibility of the framework is further demonstrated through its web-based deployment interface, which reduces the barrier to entry for non-specialist users and enables real-time threat assessment without requiring expertise in machine learning. The combination of high precision, interpretable outputs, and interactive accessibility positions this framework as a compelling candidate for integration into operational network security workflows.

While the current iteration of the system exhibits certain performance limitations, notably a constrained recall rate that merits targeted remediation, the overall results affirm the transformative potential of adaptive, intelligence-driven approaches in addressing the increasingly complex challenge of network intrusion detection.

IX. FUTURE SCOPE

Several promising avenues exist for extending and strengthening the capabilities of the proposed framework:

- Streaming data integration coupling the classification engine with real-time network telemetry feeds would enable continuous, proactive threat identification rather than retrospective batch analysis, substantially reducing the window of exposure to active intrusions.
- Scalable cloud-native deployment migrating the framework to a distributed cloud architecture would allow it to accommodate enterprise-scale traffic volumes while enabling geographically distributed security monitoring across heterogeneous network environments.
- Deep sequential modelling architectures such as Long Short-Term Memory (LSTM) recurrent networks and temporal Convolutional Neural Networks (CNNs) are well-positioned to capture the sequential and temporal dependencies inherent in network traffic streams, potentially yielding significant improvements in detection sensitivity.
- Error rate optimization dedicated efforts to reduce both false positive alerts and missed attack detections through techniques such as cost-sensitive learning, threshold calibration, and synthetic minority oversampling would meaningfully increase the system's operational reliability.
- Continual and federated learning equipping the system with mechanisms for ongoing model adaptation from new traffic observations would allow it to evolve in response to emergent threat patterns, while federated learning paradigms could enable collaborative model improvement across multiple organizational environments without compromising data privacy.

X. APPENDIX

The complete development, training, and evaluation of the proposed framework was conducted using the following open-source software stack:

- Python primary implementation language for all pipeline components
- Scikit-learn machine learning library used for classifier instantiation, training, and evaluation
- Pandas and NumPy data manipulation and numerical computation libraries underpinning the preprocessing pipeline
- Matplotlib and Seaborn visualization libraries used to generate performance plots and graphical result summaries
- SHAP attribution analysis library used to compute and visualize feature-level prediction explanations

ACKNOWLEDGEMENT

The author wishes to express heartfelt appreciation to the academic supervisors and faculty at JSPM University whose consistent mentorship and constructive evaluation shaped the intellectual trajectory of this work. Their expertise and dedication created an environment in which rigorous enquiry could flourish.

Sincere acknowledgement is also extended to the broader open-source community whose sustained collaborative effort produced the tools: Python, Scikit-learn, Pandas, NumPy, Matplotlib, and SHAP, that made this research computationally tractable. The contributors responsible for the NSL-KDD dataset deserve particular recognition for providing a methodologically sound benchmark that continues to serve as an invaluable resource for the research community.

Finally, the author is grateful to fellow researchers and colleagues whose intellectual engagement and encouragement provided both motivation and perspective throughout this undertaking.

REFERENCES

[1] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, 2009.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[3] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy*, pp. 305–316, 2010.

[4] F. Pedregosa *et al*., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[5] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, 2000.

[6] K. Kendall, "A database of computer attacks for the evaluation of intrusion detection systems," M.S. thesis,

[7] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Tech. Rep. 99-15, Dept. Comput. Eng., Chalmers Univ. Technol., 2000.