Grenze International Journal of Engineering and Technology, June Issue

CROSS-SECURE: A Domain-Adaptive IDS with Real-Time Web Interface

J. Nagaraju[1], Chegu Jahnavi Aarthi[2], Ponakala Tarsha Siva Teja[3] and Ittadi Ratna Kumar[4]
[1-4]Lakireddy Bali Reddy College of Engineering, Mylavaram, India
Email: jnrcse1@gmail.com, chegujahnaviaarthi@gmail.com, ptst2005@gmail.com, ratnakumarettadi@gmail.com

***Abstract*—In order to overcome the problems and the difficulties that modern networks are currently facing, flexibility is just as essential as algorithms. Even though intrusion detection systems (IDS) have become more effective with the inclusion of machine learning (ML) techniques, most of the models used in machine learning have an underlying major problem in that they can only work in the environment or domain in which they are trained. The aim of this study is to overcome these socalled ”domain shift” problems. However, instead of relying on traditional approaches, which cannot generalize, such as SVM or Decision Trees, we are proposing a domainadaptive model that was trained on several data sets. With the help of domain adaptation, we introduce you to CROSSSECURE, a system that not only provides robustness but allows users to visualize these insights through a real-time online interface. This research provides a solution that can be adapted to the networks of today by bridging the gap between the accuracy of models and practicality.**

***Index Terms*— Machine Learning, Cross-Domain Evaluation, Domain Adaptation, Cyber Security, Anomaly Detection, Real-Time Threat Detection, Intrusion Detection System (IDS), Network Security.**

I. INTRODUCTION

Computer networks are a vital part of modern technology, supporting communication, healthcare, finance, industry, cloud computing, mobile systems, and IoT devices. As network usage grows, security risks also increase, giving attackers more opportunities to exploit vulnerabilities. To strengthen protection, organizations must focus on areas such as threat analysis and secure network design. Cyber threats such as ransomware, malware, DDoS attacks, data breaches, and unauthorized access are becoming more advanced. Traditional tools like firewalls and signature-based intrusion detection systems mainly depend on predefined rules and known attack patterns. While effective against familiar threats, they often fail to detect new and evolving attacks in complex network environments. Machine learning-based intrusion detection systems have gained attention because they can identify hidden patterns in network traffic data. Techniques such as Random Forest, Support Vector Machine, and other traditional models have shown good results. More recently, deep learning methods like auto encoders, CNNs, and RNNs have further improved detection accuracy. However, many IDS models are trained and tested using only one dataset. As a result, their performance often drops when used in different network environments. This issue, known as domain shift, occurs when training data differs significantly from real-world deployment data. To address this challenge, this paper introduces CROSS-SECURE, a domain-adaptive intrusion detection system designed for multiple network environments.

Grenze ID: 01.GIJET.12.2.518_1

It uses benchmark datasets for training and applies a Domain-Adversarial Neural Network to reduce dataset bias. The system supports real-time flow-based intrusion detection, improves robustness, and provides monitoring through a web-based interface, making IDS deployment more practical and effective.

II. Related Work

Intrusion Detection Systems (IDS) have long been important in network security due to increasingly advanced cyber threats. Early IDS models were signature-based, using known attack patterns. While effective for known threats, they could not detect new attacks. Machine learning methods such as SVM, Decision Tree, Naıve Bayes, KNN, and Random Forest improved IDS performance by identifying traffic patterns. However, their success depended heavily on training data quality. Datasets like KDD Cup 99 and NSL-KDD sup- ported early research but later became outdated. Newer datasets such as CICIDS-2017 and UNSW- NB15 better represent modern traffic and attacks, though many studies still rely on only one dataset. Deep learning models such as auto encoders, RNNs, and CNNs further improved detection by learning features automatically. However, they often require high computational resources. A major issue is domain shift, where models trained on one dataset perform poorly on others. Domain-Adversarial Neural Networks (DANN) help solve this by learning features that generalize across datasets. The CROSS-SECURE model combines deep learning, domain adaptation, machine learning, and a real-time web dashboard to improve intrusion detection accuracy and practical deployment.

TABLE I. RECENT CROSS-DOMAIN IDS LITERATURE

Paper	Dataset(s)	Limitations	Future Work
Zhang et al. (2021)	NSL-KDD, CICIDS-2017	Poor unseen-domain performance	Multi-domain adaptation
Li et al. (2020)	NSL-KDD, UNSW-NB15	No real-time validation	Live traffic integration
Ferrag et al. (2021) Yang et al. (2022)	CICIDS-2017 CICIDS-2017, UNSW-NB15	Single-dataset testing High computation cost	Cross-dataset evaluation Lightweight IDS models
Kim et al. (2022)	NSL-KDD, UNSW-NB15	Resource-heavy training	Hybrid adaptive models
Zhou et al. (2022) Chen et al. (2023)	CICIDS-2017 NSL KDD, CICIDS-2017, UNSW-NB15	Limited live scalability No alert visualization	Flow-level optimization Web-based deployment
Alshamrani et al. (2023)	UNSW-NB15, CICIDS-2017	Offline evaluation only	Live system integration

III. Proposed Methodology

The goal of the planned CROSS-SECURE method is to develop a reliable and domain-adaptive intrusion detection system (IDS) that can potentially support real-time detection through a web interface and is also generalized across a variety of network environments. Data preprocessing, feature alignment, machine learning, deep learning, domain adaptation, flow-based detection, and visualization through a web interface are some of the techniques in the proposed method's framework.

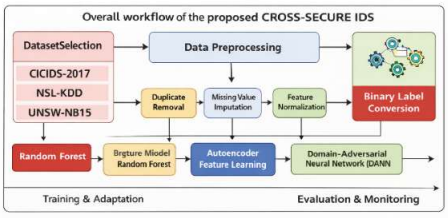

Figure 1. Overall workflow of the CROSS-SECURE IDS

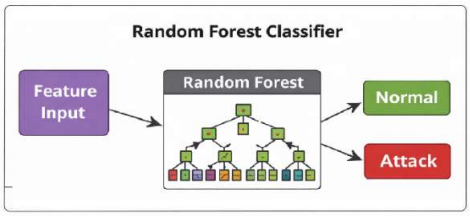

Figure 2. Random Forest-based baseline intrusion detection

The image shows the whole process, which involves preprocessing and collecting the dataset then going through the process of features learning and domain adaptation, followed by the real-time detection and web-based monitoring.

A. Stage 1: Dataset Selection

In order to ensure the variability of the network response, as well as the variety of attacks, three most popular benchmark IDS databases are considered. Therefore, due to the presence of realistic and modern network traffic, as well as different types of attacks, CICIDS-2017 is considered for application in modern IDS research. Due to the consideration of hybrid and newly created attacks, UNSW-NB15 is used as the undetected test domain.

B. Stage 2: Data Preprocessing

In order to ensure data reliability and data consistency, a preprocessing step is carried out methodically on each dataset. For avoiding biases that may occur during the training of models, duplicate and redundant data is removed.

For ensuring data integrity, statistical imputation methods are employed to deal with missing data. In this case, NaN is used to replace infinite data. Then these infinite data values are cleansed. The correct encoding schemes are employed to convert data into numerical format. In the case of numerical data, Standard Scaler is employed. Binary classes, such as normal and attack, are created from multi-class attack classes.

C. Stage 3: Feature Space Unification

As the feature sets for each dataset are different, some common network-related features are chosen to build a single feature space from the three datasets. Flow Duration, Source Bytes, Destination Bytes are some of the chosen unified feature sets. To build a uniform input format for the model, feature mapping is done for these sets among all the datasets.

D. Stage 4: Feature Standardization

A dataset of CICIDS-2017 is used to train a StandardScaler, and this will aid in making the model stable by avoiding bias in features with larger numerical ranges. Then, when the model is used in real-time detection, the above scaler will be used again to scale the network flows in real time.

E. Stage 5: Baseline Classification using Random Forest

A Random Forest classifier is trained by using those features which are provided by CICIDS-2017 after preprocessing and scaling. This classifier has been selected due to its high accuracy and low chances of overfitting. It is also robust to noise and has potential to show nonlinear correlations. Hence, this classifier would be preferred as a baseline IDS.

F. Stage 6: Latent Feature Extraction

The CICIDS-2017 dataset is utilised for training an auto encoder, which enhances the representation of features and reduces noise.

To identify essential patterns in network traffic, the Auto encoder com- presses and reconstructs input features into a latent space.

G. Stage 7: Adversarial Domain Adaptation

To overcome the domain shift, the latent at- tributes are analyzed through the use of a Domain- Adversarial Neural Network (DANN). It consists of a feature extractor, an attack classifier, and a domain classifier. The Gradient Reversal Layer (GRL) ensures domain-invariant learning.

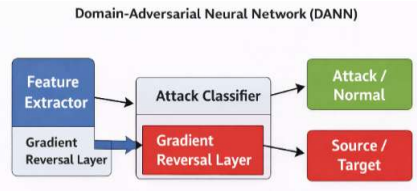

Figure 3. Domain-Adversarial Neural Network (DANN) architecture

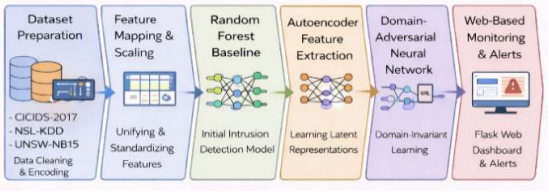

Figure 4. Flowchart of the Implementation Pipeline of CROSS-SECURE IDS

H. Stage 8: Cross-Domain Validation

With the newly unexplored UNSW-NB15 dataset, we tested the trained DANN to comprehensively validate the model's ability to generalize. Accuracy, precision, recall, F1-score, and confusion matrix analysis are just some of the numerous metrics employed to assess performance in ensuring the results are reliable and valid.

I. Stage 9: Real-Time Traffic Analysis

We developed a stream processing engine for real-time monitoring using the Scapy library. By assembling raw packets into distinct network flows based on the standard 5-tuple identifier (Source/Dest IP, Source/Dest Port, Protocol), this module calculates essential parameters in real-time for immediate classification, such as flow size and bytes.

IV. IMPLEMENTATION

The proposed Intrusion Detection System (IDS) combines deep learning, machine learning, and web technologies to deliver effective intrusion detection with real-time monitoring. The process begins with preprocessing two bench- mark datasets: UNSW-NB15 and CICIDS-2017. This includes converting categorical data into numerical form, handling missing values, removing duplicates, and normalizing numerical features using Standard- Scaler for consistent scaling. After preprocessing, the CICIDS-2017 dataset is used to train an auto encoder. The auto encoder reduces noise, learns hidden relationships among network features, and creates a compact feature representation. These encoded features are then used to train a Random Forest classifier as the baseline detection model.

To improve performance across different datasets, a Domain-Adversarial Neural Network (DANN) is applied. This model includes a feature extractor, attack classifier, and domain classifier. A gradient reversal layer helps generate domain-invariant features, improving generalization. The trained model is then tested on the original UNSW-NB15 dataset without modification to evaluate cross-domain performance. Standard metrics such as accuracy, precision, recall, F1-score, and confusion matrix are used for assessment. For practical deployment, a lightweight web interface is developed using HTML, CSS, JavaScript, and Flask. The system provides real-time predictions, accepts network traffic samples, generates intrusion alerts, and stores attack logs in a database.

V. RESULTS

The UNSW-NB15 and CICIDS-2017 benchmark datasets were used to evaluate the proposed frame- work. The main goals were to test the real-time monitoring dashboard and measure cross-dataset generalization through domain adaptation. First, a baseline Random Forest (RF) model was trained on CICIDS-2017 features. When tested on the unseen UNSW-NB15 dataset, it achieved only 40% accuracy, showing the impact of domain shift between training and testing environments.

To solve this issue, a two-stage framework was introduced. An auto encoder first extracted compact and noise-free features, followed by a Domain- Adversarial Neural Network (DANN) to learn domain-invariant patterns across datasets. This method improved performance significantly, achieving about 65% accuracy on UNSW-NB15 and outperforming the baseline model.

The results show that adversarial domain adaptation can build more robust IDS models for multiple environments. Figure 5 visually compares the results, showing that the DANN-based model remains stable in unseen environments, while the baseline model is affected by domain mismatch. A complete web interface was also developed using HTML/JavaScript for the frontend and Flask for the backend. As shown in Figure 6, the dashboard provides real-time network monitoring with details such as timestamps, IP addresses, flow duration, and color-coded threat alerts. All alerts are stored in the Firebase database for future analysis.

TABLE II. RECENT CROSS-DOMAIN IDS LITERATURE

Model	Accuracy (%)	Precision	Recall	F1
Random Forest (Baseline)	40	0.38	0.42	0.4
Proposed Auto encoder + DANN	65	0.64	0.66	0.65

In conclusion, the data from the experiment shows how CROSS-SECURE closes the gap between the design of the algorithm and the practical application of security monitoring in real-time, as well as outperforming the traditional machine learning algorithms in the cross-domain setting.

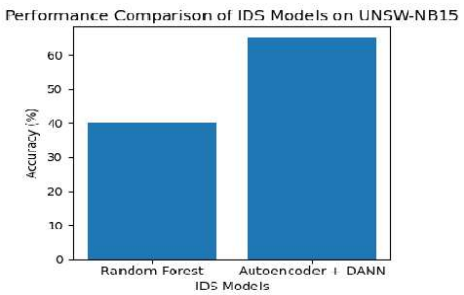

Figure 5. Accuracy comparison of Random Forest and Auto encoder +DANN

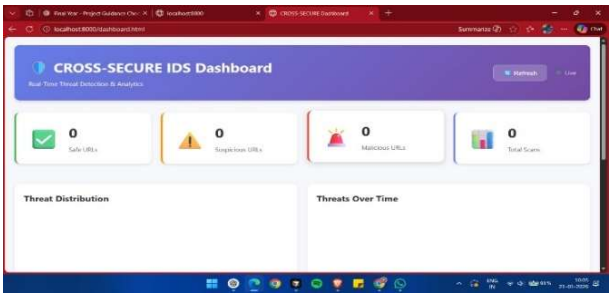

Figure 6. Real-time IDS web interface displaying live intrusion alerts

VI. Conclusion and Future Work

This study addressed the limited adaptability of many machine learning models in network security. To overcome this issue, we developed CROSS- SECURE, an intrusion detection system that com- bines domain-adaptive deep learning with a real- time monitoring interface. Using multiple datasets, auto encoders, and a Domain-Adversarial Neural Network (DANN), the system reduced domain shift and improved cross-environment detection.

Results showed that a traditional Random Forest model achieved 40% accuracy on unseen data, while the proposed framework improved accuracy to 65%. The addition of a web-based dashboard and flow- based detection engine also made the system more practical for security operators. Future work will include testing with more datasets such as TON IoT and CSE-CIC-IDS2018, and exploring advanced models like Transformers and Graph Neural Networks (GNNs). The system will also be extended to multi-class classification for identifying specific attack types, along with high- speed traffic analysis and automated threat blocking.

References

[1] D. E. Denning,"Anintrusion-detection model," IEEE Trans. Software Engineering, vol. SE-13, no. 2, pp. 222–232, 1987.

[2] KDD Cup 1999 Data, "The Third International Knowledge Discovery and Data Mining Tools Competition," 1999.

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

[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. ICISSP, 2018.

[5] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. MilCIS, 2015.

[6] M. Ring et al.,"Asurvey of network-based intrusion detection data sets," Computers & Security, vol. 86, pp. 147–167, 2019.

[7] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," Cybersecurity, vol. 2, no. 20, 2019.

[8] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[9] A. L. Buczak and E. Guven,"Asurvey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2016. [7] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.

[10] R. Vinayakumar et al., "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE TPAMI, vol. 35, no. 8, pp. 1798–1828, 2013.

[12] N. Shone et al.,"Adeep learning approach to network intrusion detection," IEEE Trans. Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, 2018.

[13] W. Wang et al., "HAST-IDS: Learning hierarchical spatial–temporal features using deep neural networks," IEEE Access, vol. 6, pp. 1792–1806, 2018.

[14] M. Al-Jarrah et al., "Network anomaly detection using supervised machine learning," Future Generation Computer Systems, vol. 78, pp. 16–28, 2018.

[15] T. T. Nguyen and G. Armitage, "Asurvey of techniques for Internet traffic classification," IEEE Commun. Surveys Tuts., vol. 10, no. 4, pp. 56–76, 2008.

[16] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor traffic using time-based features," in Proc. ICISSP, 2017, pp. 253–262.

[17] S. Revathi and A. Malathi, "A detailed analysis on NSL- KDD dataset using various machine learning techniques for intrusion detection," Int. J. Eng. Res. Technol., vol. 2, no. 12, pp. 1848–1853, 2013.

[18] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," Computers & Security, vol. 98, 2020.

[19] J. Zhang, Y. Chen, and X. Chen, "Cross-domain network intrusion detection based on deep transfer learning," IEEE Access, vol. 9, pp. 128721–128734, 2021.

[20] A. Javaid, Q. Niyaz, W. Sun, and M. Alam,"Adeep learning approach for network intrusion detection system," in Proc. IEEE Biometric Comp., 2016, pp. 21–26.