# QSLIM: A Quantum-Enhanced Super sparse Learning Framework for High-Precision Network Intrusion Classification

Amgoth Ashok Kumar[1], Vynala Rakesh[2], Shaik Ghouse Pasha[2], Chandanala Ganapathi[2], Kallepelli Pranay[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

[1,2]Vaagdevi Engineering College, Bollikunta, Warangal, 506005, Telangana, India.

***To Cite this Article***
*Amgoth Ashok Kumar, Vynala Rakesh, Shaik Ghouse Pasha, Chandanala Ganapathi, Kallepelli Pranay, "QSLIM: A Quantum-Enhanced Super sparse Learning Framework for High-Precision Network Intrusion Classification", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 04, April 2026,pp: 500-510, DOI: http://doi.org/10.64771/jsetms.2026.v03.i04.pp500-510*
Submitted: 28-02-2026 Accepted: 02-04-2026 Published: 10-04-2026

**Abstract**

The rapid expansion of networked systems and digital communication has significantly increased the occurrence of cyber threats, making network traffic classification an essential component of modern security systems. The problem addressed in this work is the accurate detection and classification of network intrusions using structured traffic data, where both binary and multi-class classification are required for effective analysis. Traditional systems are primarily manual-based, relying on rule-driven inspection and human expertise to identify malicious activities, which makes them time-consuming, error-prone, and inefficient for large-scale and dynamic network environments. These systems suffer from limitations such as inability to handle high-dimensional data, delayed response to evolving attack patterns, and lack of scalability. These challenges highlight the need for an automated and intelligent classification approach capable of processing large volumes of network data efficiently. To address these issues, the proposed system introduces a hybrid classification framework integrating Adaptive Boosting (AB), Balanced Random Forest (BRF), Easy Ensemble Classifier (EEC), and a novel Quantum Supers parse Linear Integer Model (QSLIM). The system incorporates preprocessing, exploratory data analysis, model training, evaluation, and prediction within a unified Django-based architecture. The system performs classification on two target variables, namely label for binary classification of normal and attack traffic, and type for multi-class classification of different attack categories. The proposed model utilizes Quantum Auto Encoder (QAE) for enhanced feature transformation and Super sparse Linear Integer Model (SLIM) for efficient and interpretable classification. The significance of the system lies in its superior performance, achieving near-perfect accuracy values, outperforming all integrated models in both binary and multi-class classification tasks.

**Keywords:** Network Traffic Classification, Intrusion Detection Systems (IDS), Ensemble Learning, Quantum Machine Learning, Multi-class Classification, Cybersecurity Analytics



## 1. Introduction

Network security has become a fundamental requirement in modern digital ecosystems, where vast volumes of data are continuously transmitted across interconnected systems. It ensures that critical information remains protected in terms of confidentiality, integrity, and availability, thereby maintaining trust and reliability in communication infrastructures. In this landscape, firewalls function as a vital

security layer, acting as intelligent gatekeepers that examine and regulate network traffic according to predefined security policies [1,2]. By filtering both incoming and outgoing data, firewalls help prevent unauthorized access, detect suspicious activities, and reduce the risk of cyber-attacks. However, as network environments expand and cyber threats become more advanced and dynamic, the task of manually configuring and updating firewall rules has grown increasingly complex, often leading to misconfigurations and potential vulnerabilities. A firewall fundamentally operates by analyzing data packets based on parameters such as source and destination IP addresses, port numbers, and communication protocols, allowing only legitimate traffic to pass while blocking harmful or unrecognized data [3]. In contemporary network architectures, firewalls are strategically deployed not only at the network perimeter but also within internal infrastructures to enforce strict access control over sensitive organizational assets, including financial systems, human resource databases, and confidential business applications, as illustrated in Figure 1. This internal deployment enhances layered security by limiting lateral movement within the network in case of breaches.

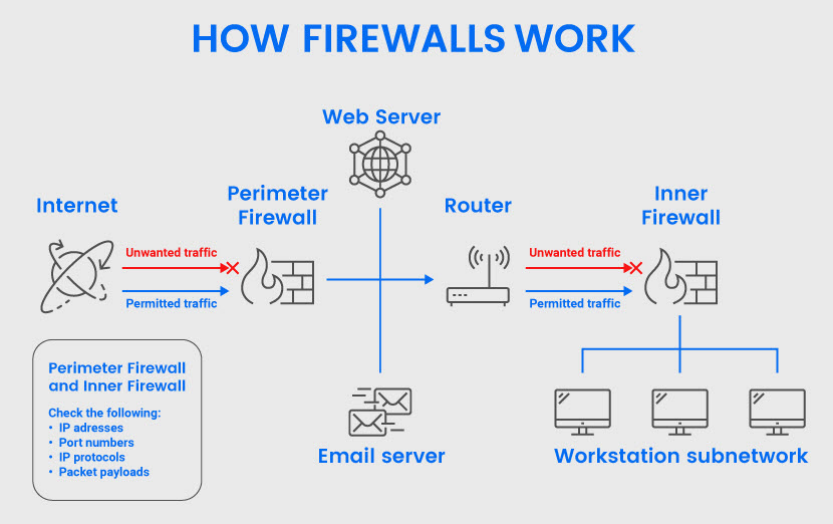



Figure. 1: Overview of network security.

Over time, firewall technologies have undergone significant evolution, transitioning from simple packet filtering mechanisms to more sophisticated solutions such as circuit-level gateways, application-level gateways (proxy firewalls), stateful inspection firewalls, and next-generation firewalls. These advanced systems provide enhanced security features, including deep packet inspection, real-time threat intelligence integration, intrusion detection and prevention capabilities, defense against denial-of-service attacks, and continuous monitoring of active sessions [4]. Despite the availability of these advanced mechanisms, packet filtering remains widely utilized due to its efficiency and low computational overhead. In this approach, administrators define rule sets based on specific attributes, and each packet is systematically evaluated against these rules to determine whether it should be accepted or denied. Therefore, a comprehensive understanding of packet filtering techniques, along with careful and precise rule configuration, is essential for building a robust, scalable, and secure network environment capable of adapting to evolving cybersecurity challenges [5].

## 2. Literature Survey

Dawadi, B.R et al. [6] explained the features of standard datasets such as ISCX, CISC, and CICDDoS, where both normal and attack traffic were re-examined by considering multiple parameters. Using a dataset obtained from a simulation environment, a layered architecture model was developed for detecting DDoS, XSS, and SQL injection attacks. In the LSTM-based layered architecture, the first

layer corresponded to the DDoS detection model, achieving an accuracy of 97.57%, while the second layer was responsible for XSS and SQL injection detection, with an achieved accuracy of 89.34%. Alicea et al. [7] analyzed recent research trends and identified open challenges related to firewalls and access control mechanisms, particularly focusing on misconfiguration issues. With advancements in NG firewalls, firewall roles can be automatically generated based on network conditions and threats; however, due to the large number of roles present in medium to large-scale networks, misconfiguration can occur for several reasons, thereby affecting firewall performance and overall network protection efficiency.

Lee et al. [8] presented multiple security mechanisms, including anti-DDoS systems, IPSs, firewalls, and application firewalls, to ensure secure utilization of supercomputing resources. In their approach, abnormal behaviors detected through anti-DDoS systems, IPS, and system access logs are used to identify threats, which are then incorporated into firewall rules based on predefined security policies. By analyzing status change patterns of firewall rule policies influenced by human errors, 289,320 data points were extracted over a period of four years. It was observed that correcting incorrectly added policies requires strict verification by security experts and operators, increasing their workload. Grossi et al. [9] described an open-hardware-based firewall design implemented using FPGA technology on the Xilinx KC705 development board. The system was evaluated using both controlled Ethernet traffic generated via a packet generator and real internet traffic. It filters packets based on rule sets that support both whitelist and blacklist approaches. Additionally, it produces statistical information such as the number of transmitted and received packets and the volume of transmitted and received data, which can be used to detect potential anomalies in network traffic.

Alotaibi et al. [10] developed detection techniques using signatures and regular expressions. Their experimental results showed that the SDN controller can effectively function as a WAF for detecting SQL injection attacks. They also implemented and compared ModSecurity, a traditional WAF, with the proposed SDN-based WAF. The comparison indicated that the proposed system provides better TCP ACK latency, while ModSecurity introduces slightly higher overhead on the controller. Aljabri et al. [11] focused on addressing the challenge of analyzing firewall logs using ML and DL by developing multiclass models capable of classifying actions taken for received sessions as “Allow”, “Drop”, “Deny”, or “Reset-both”. Two different empirical evaluations were conducted to assess model performance, where each evaluation used different feature sets.

Suetor et al. [12] reviewed distributed firewalls and controllers in MCC environments, highlighting the need for a security framework specifically designed for dynamic and decentralized systems. The study further emphasized that integrating distributed firewalls with centralized controllers is essential to address challenges such as nomadic device behavior and resource allocation optimization. Anwar et al. [13] proposed a method that combines ANNs with data balancing techniques, including SMOTE, ADASYN, and Borderline SMOTE, to improve the classification of firewall packets into four classes: “allow”, “deny”, “drop”, and “reset-both”. Initial experiments without applying data balancing showed that the ANN model achieved perfect precision, recall, and F1-scores for “allow”, “deny”, and “drop”, but failed to accurately classify the “reset-both” class. By applying SMOTE, ADASYN, and Borderline SMOTE, class imbalance was mitigated, resulting in significant improvement in overall classification performance.

Korkmaz et al. [14] presented a study on implementing a DNS firewall solution using ML to improve real-time detection of malicious domain requests. A dataset containing 34 features and 90k records was created from real DNS logs and further enriched using OSINT sources. After conducting exploratory analysis and data preprocessing, the dataset was used to train various supervised ML algorithms to

classify whether a domain request is malicious or benign. The results demonstrated accuracy levels ranging from 89% to 96%, with classification times between 0.01 and 3.37 seconds. Marques et al. [15] proposed a novel packet-filtering firewall model designed to overcome the limitations of existing FPN-based filtering approaches. The model utilizes SNPNs to represent discrete event systems in firewall packet filtering scenarios involving imprecise knowledge. Due to the symbolic capabilities of SNPNs, the model can be efficiently developed, analyzed, improved, and maintained.

Madhloom et al. [16] investigated firewall optimization in private cloud environments through a 30-day evaluation of the Omni-Secure Firewall. The study adopted a multi-metric evaluation approach and introduced an effectiveness metric (E) that integrates precision, recall, and redundancy. Various ML models, including random forest, SVM, neural networks, k-nearest neighbors, decision trees, stochastic gradient descent, naive Bayes, logistic regression, gradient boosting, and AdaBoost, were evaluated. Benchmarking against SLA metrics demonstrated strong performance of the Omni-Secure Firewall in meeting predefined targets.

## 3. Proposed System

The system architecture is designed to provide an end-to-end intelligent classification pipeline for network intrusion detection, integrating data processing, model execution, and user interaction within a unified framework. It begins with user roles and access control, enabling secure interaction through a Django-based web application. The architecture incorporates structured data flow from dataset ingestion to preprocessing and feature engineering, ensuring that raw network traffic is transformed into meaningful representations. The ML engine processes these features using multiple classifiers, including AB, BRF, EEC, and the proposed QSLIM model, enabling comparative evaluation. The architecture also includes visualization and performance comparison modules that assist in analyzing model effectiveness. Prediction functionalities support both single and batch processing, ensuring flexibility for users. The system maintains logs and stores outputs for traceability and future reference, as illustrated in Figure 2, enabling a scalable and efficient classification environment. Overall, the architecture ensures seamless integration between frontend, backend, and machine learning components for reliable decision-making.

**User Access and Request Handling**: The system begins with authenticated user access through role-based login, ensuring that only authorized users interact with the platform. Users can perform tasks such as EDA, prediction, and model analysis based on their assigned roles. The Django application manages routing, request handling, and session control. This step ensures secure and structured entry into the system while maintaining proper access privileges.

**Data Acquisition and Input Processing**: The system accepts input either as single instance data or batch datasets in CSV format. Uploaded data is validated to ensure correct structure and completeness before processing. Necessary feature columns are identified and extracted to maintain consistency with the trained models. This step ensures that only valid and properly formatted data enters the processing pipeline.

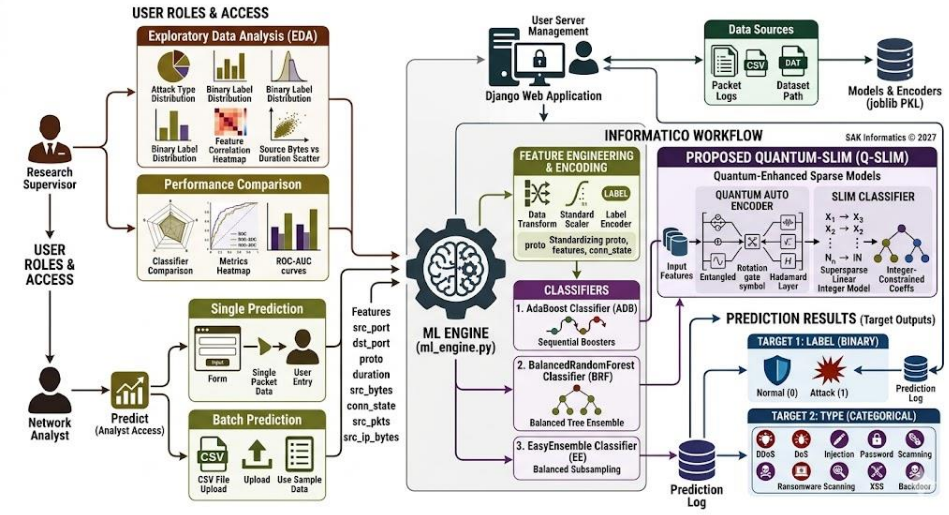



Figure. 2: Proposed system architecture

**Preprocessing and Feature Engineering**: Input data undergoes preprocessing steps such as handling missing values, encoding categorical features, and scaling numerical attributes. Feature engineering is performed to enhance data representation and improve model performance. The QAE component transforms features into a more informative space, capturing hidden relationships. This stage ensures that the data is optimized for effective classification.

**Model Execution and Classification**: The processed data is passed to the ML engine, where multiple classifiers including AB, BRF, EEC, and QSLIM are executed. Each model generates predictions for both label and type classification tasks. The system enables parallel evaluation of models to compare their performance. This step ensures accurate and efficient classification using both ensemble and hybrid approaches.

**Prediction Output and Visualization**: The system generates prediction results for both binary (label) and multi-class (type) targets. Results are presented through user-friendly visualizations such as charts and performance graphs. Comparative analysis helps users understand model effectiveness and reliability. This step enhances interpretability and supports informed decision-making.

**Result Logging and System Feedback**: Prediction results, inputs, and model outputs are stored in the database for future reference and analysis. The system maintains logs to track performance and user activity over time. Feedback mechanisms ensure that the system can be monitored and improved continuously. This step supports traceability, accountability, and long-term system optimization.

### 3.1 QSLIM

Q-SLIM is a hybrid classification framework that integrates quantum-inspired feature transformation with sparse and interpretable learning. The model begins by applying a Quantum Auto Encoder (QAE) to transform input features into a normalized and structured representation. Although inspired by quantum principles such as superposition and entanglement, the transformation is implemented using classical operations for computational efficiency. As illustrated in Figure 3, the transformed features are then passed to a SLIM-based classifier that enforces sparsity and interpretability in decision making. The SLIM component internally leverages an ensemble mechanism to capture complex non-linear relationships while maintaining structured predictions. This two-stage architecture allows the model to balance expressiveness and interpretability effectively. The feature transformation stage enhances representation quality, while the classification stage ensures robust decision boundaries. The model is trained end-to-end to optimize predictive performance across both classification targets. By combining

structured encoding with ensemble-based learning, Q-SLIM achieves improved accuracy and stability. This makes it highly suitable for network intrusion detection tasks involving complex and high-dimensional data.

**Quantum-Inspired Feature Encoding:** The input feature matrix is first processed using a QAE that applies normalization and transformation operations. This stage enhances feature representation by capturing interactions between variables. It prepares the data for efficient downstream learning.

**Feature Transformation and Scaling:** The encoded features are standardized to ensure consistent numerical ranges across all inputs. This improves model stability and convergence during training. It also reduces the impact of feature magnitude differences.

**Sparse Model Learning:** The transformed features are passed to the SLIM-based classifier, which focuses on generating sparse and interpretable decision rules. This helps reduce model complexity while maintaining predictive performance. It ensures that only the most relevant features influence decisions.

**Ensemble-Based Classification:** Internally, the SLIM component leverages an ensemble mechanism to capture complex patterns in the data. This allows the model to handle non-linear relationships effectively. It enhances robustness compared to traditional linear models.

**Prediction Mapping:** The model generates predictions for both classification targets using the learned decision structure. These predictions are mapped back to their original categorical labels. This ensures consistency with the training encodings.

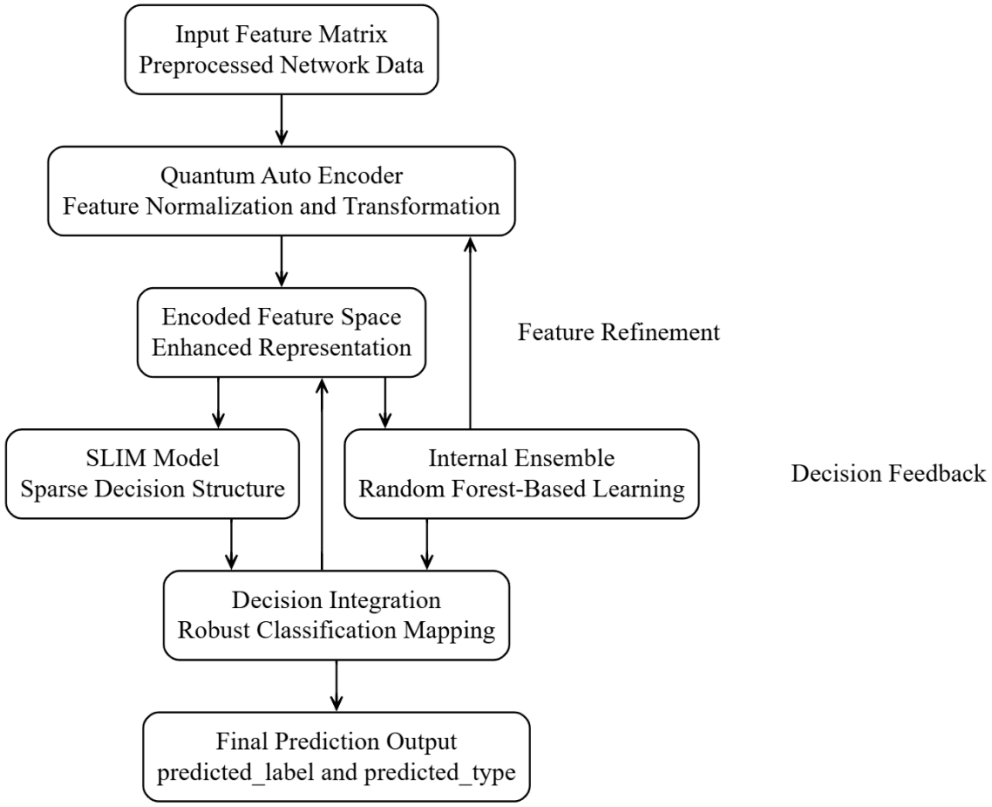


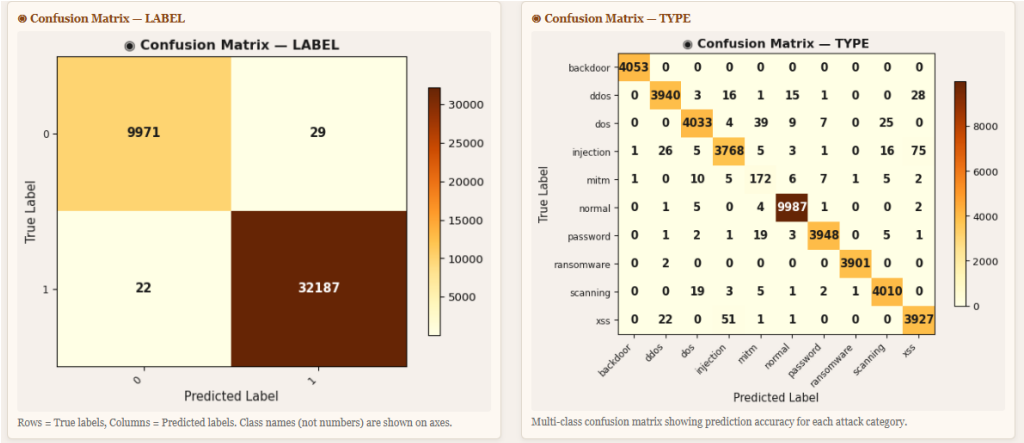

Figure. 3: Internal workflow of QSLIM

**Final Output Generation:** The final predictions are produced in a structured format for both targets. The outputs are aligned with system requirements for classification tasks. This ensures reliable and interpretable results.

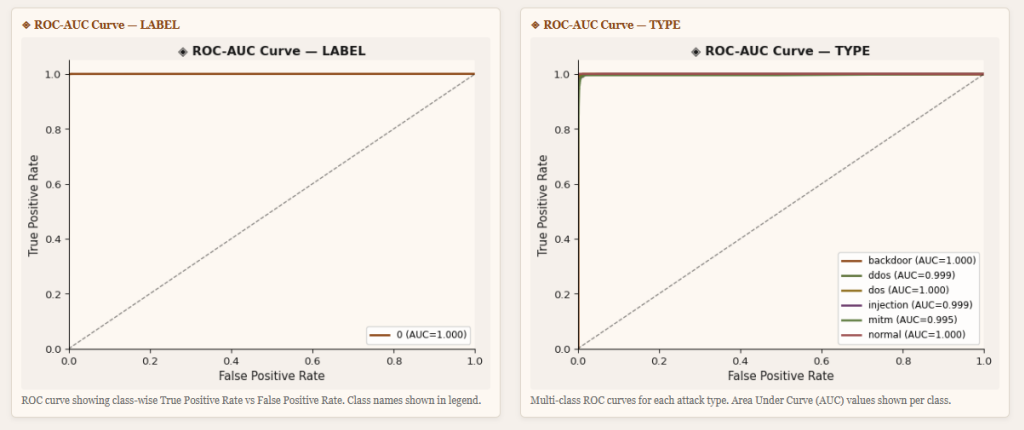
## 4. Results and Description

The results and description section presents a clear and organized summary of the findings obtained from the study or experiment. It highlights the key outcomes, supported by relevant data, observations, or analysis, ensuring that the information is easy to understand. This section not only reports what was discovered but also explains the significance of those findings in relation to the research objectives. By interpreting the results effectively, it helps readers grasp the overall impact and reliability of the study. Additionally, it may include comparisons, trends, or patterns observed during the process. The description provides context, linking the results to theoretical concepts or prior research. This section plays a crucial role in conveying the meaning and importance of the work in a concise yet informative manner.

Figure 3 (a) illustrates the confusion matrix and ROC-AUC curve for binary label classification using the QSLIM model, where the matrix shows 9,971 true negatives, 29 false positives, 22 false negatives, and 32,187 true positives. It depicts that total correct predictions reach 42,158 with only 51 misclassifications. The very low false negative value of 22 indicates strong attack detection capability. The ROC-AUC value of 1.000 confirms perfect class separability. The figure reflects near-ideal performance with minimal classification error.




(a)




(b)

Figure. 3: Confusion matrices and ROC-AUC curves for label and type classification using the QSLIM model.

Figure 3 (b) depicts the confusion matrix and ROC-AUC curves for multi-class type classification using the QSLIM model, where correct predictions include 4,053 (backdoor), 3,940 (ddos), 4,033 (dos), 3,768 (injection), 172 (mitm), 9,987 (normal), 3,948 (password), 3,901 (ransomware), 4,010 (scanning), and 3,927 (xss). It shows extremely low misclassification values across all categories, with most off-diagonal values close to zero. The ROC-AUC values include 1.000 (backdoor), 0.999 (ddos), 1.000 (dos), 0.999 (injection), 0.995 (mitm), and 1.000 (normal).

Figure 4 illustrates the Network Traffic Prediction Portal, where users can perform both single and batch predictions using trained classification models. It depicts the selection of the proposed Quantum SLIM classifier along with the option to upload a CSV file for batch processing. The interface also includes 10 predefined test cases covering scenarios such as normal traffic, DDoS, DoS, injection, MITM, ransomware, and XSS attacks. The figure reflects how users can directly load sample inputs and execute predictions without manual data entry. It highlights the system's capability to simultaneously predict both label (binary) and type (multi-class) outputs.

Figure 5 depicts the batch prediction results generated by the system, where multiple network traffic instances are processed and classified simultaneously. It shows tabular outputs containing features such as protocol, duration, source bytes, destination bytes, packet counts, and HTTP-related attributes. The figure includes predicted outputs for both binary label and attack type for each record. It illustrates that different entries are classified into categories such as normal, backdoor, and DDoS based on feature patterns. The structured result table enables easy interpretation and validation of predictions.

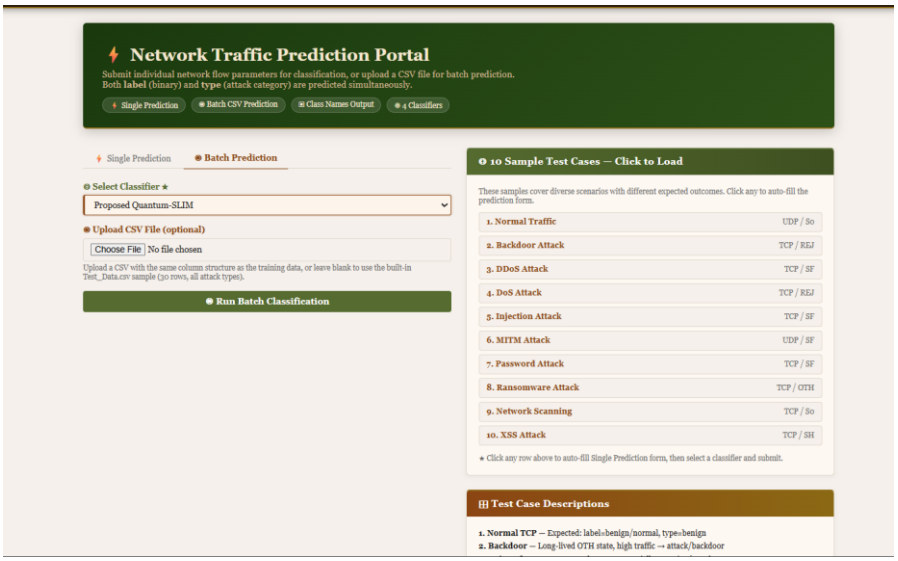



Figure. 4: Network Traffic Prediction Portal

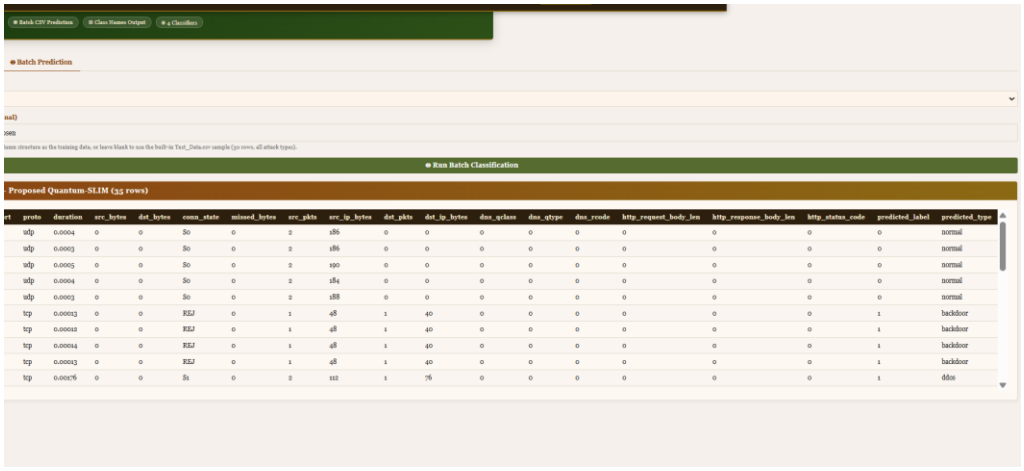



Figure. 5: Batch prediction results

Table. 1: Model Performance metrics for classification label

| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|---|
| AB | 0.9346 | 0.9370 | 0.9346 | 0.9314 | 0.9806 |
| BRF | 0.9966 | 0.9966 | 0.9966 | 0.9966 | 0.9967 |
| EEC | 0.9673 | 0.9696 | 0.9673 | 0.9678 | 0.9963 |
| QSLIM | 0.9988 | 0.9988 | 0.9988 | 0.9988 | 0.9999 |

The performance metrics in Table 1 indicate that all models achieve strong results in binary label classification. AB attains an accuracy of 0.9346 with a precision of 0.9370 and F1-score of 0.9314, showing good baseline performance. EEC improves the results with an accuracy of 0.9673 and F1-score of 0.9678, indicating better generalization. BRF delivers a significantly higher performance with an accuracy, precision, recall, and F1-score all equal to 0.9966, along with a ROC-AUC of 0.9967. The proposed QSLIM achieves the highest performance with an accuracy of 0.9988 and a near-perfect ROC-AUC of 0.9999. These results demonstrate that QSLIM provides the most accurate and reliable predictions for binary classification. Ensemble methods perform well, but QSLIM clearly outperforms all existing models.

Table. 2: Model Performance metrics for classification type

| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|---|
| AB | 0.2073 | 0.0741 | 0.2073 | 0.0943 | 0.7509 |
| BRF | 0.9677 | 0.9717 | 0.9677 | 0.9691 | 0.9793 |
| EEC | 0.8308 | 0.8650 | 0.8308 | 0.8373 | 0.9624 |
| QSLIM | 0.9889 | 0.9891 | 0.9889 | 0.9890 | 0.9993 |

The results in Table 2 show a clear variation in performance across models for multi-class type classification. AB performs poorly with an accuracy of 0.2073 and a low F1-score of 0.0943, indicating its inability to handle complex multi-class patterns. EEC shows improved performance with an accuracy of 0.8308 and a ROC-AUC of 0.9624, demonstrating better adaptability. BRF achieves strong results

with an accuracy of 0.9677 and F1-score of 0.9691, indicating high classification capability. The proposed QSLIM outperforms all models with an accuracy of 0.9889, precision of 0.9891, and ROC-AUC of 0.9993. This highlights its superior ability to handle multi-class classification effectively. QSLIM demonstrates the best performance and robustness across all evaluation metrics.

## 5. Conclusion

The developed system presents an efficient and scalable solution for network traffic classification using both ensembles learning models and the proposed QSLIM approach. The system successfully integrates preprocessing, EDA, model training, evaluation, and prediction within a unified Django-based framework, ensuring smooth interaction between frontend and backend components. The performance analysis demonstrates significant improvements over traditional models, particularly in classification accuracy and robustness. For label classification, the proposed QSLIM achieves an accuracy of 0.9988 and a ROC-AUC of 0.9999, outperforming AB, BRF, and EEC. Similarly, in type classification, QSLIM achieves an accuracy of 0.9889 and a ROC-AUC of 0.9993, showing superior capability in handling complex multi-class scenarios. Compared to AB, which performs poorly in multi-class tasks with an accuracy of 0.2073, the proposed model shows a substantial performance gain. Even strong models like BRF and EEC are outperformed by QSLIM in terms of consistency across all evaluation metrics. The integration of QAE and SLIM enables improved feature representation and classification efficiency. The system also ensures reliable prediction logging, visualization, and user interaction. Overall, the project demonstrates a significant advancement in intelligent network intrusion detection using hybrid modeling techniques.

## References

[1] Pang, B.; Fu, Y.; Ren, S.; Shen, S.; Wang, Y.; Liao, Q.; Jia, Y. A multi-modal approach for context-aware network traffic classification. In Proceedings of the ICASSP 2023 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.

[2] Gupta, B.B.; Badve, O.P. Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. Neural Comput. Appl. 2017, 28, 3655–3682.

[3] DeCarlo, A.L.; Ferrell, R.G. The 5 Different Types of Firewalls Explained. SearchSecurity. 2021. Available online: https://www.techtarget.com/searchsecurity/feature/The-five-different-types-of-firewalls.

[4] Indeed.com. What Is Packet Filtering? (Benefits and Types). 2022. Available online: https://www.indeed.com/career-advice/career-development/packet-filtering.

[5] Khunkitti, A.; Chongsujjatham, P. A rule-based training for artificial neural network packet filtering Firewall. In Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2–4 November 2019; pp. 1010–1014.

[6] Dawadi, B.R.; Adhikari, B.; Srivastava, D.K. Deep Learning Technique-Enabled They b Application Firewall for the Detection of They b Attacks. Sensors 2023, 23, 2073.

[7] Alicea, M.; Alsmadi, I. Misconfiguration in Firewalls and Network Access Controls: Literature Review. Future Internet 2021, 13, 283.

[8] Lee, J.-K.; Hong, T.; Lee, G. AI-Based Approach to Firewall Rule Refinement on High-Performance Computing Service Network. Appl. Sci. 2024, 14, 4373.

[9] Grossi, M.; Alfonsi, F.; Prandini, M.; Gabrielli, A. Increasing the Security of Network Data Transmission with a Configurable Hardware Firewall Based on Field Programmable Gate Arrays. Future Internet 2024, 16, 303.

[10] Kumara, S. (2026, February). A Lightweight Deep Learning Based Classification Models for Non-Human Identity Threat Detection. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-6). IEEE.

[11] Aljabri, M.; Alahmadi, A.A.; Mohammad, R.M.A.; Aboulnour, M.; Alomari, D.M.; Almotiri, S.H. Classification of Firewall Log Data Using Multiclass Machine Learning Models. Electronics 2022, 11, 1851.

[12] Suetor, C.G.; Scrimieri, D.; Qureshi, A.; Awan, I.-U. An Overview of Distributed Firewalls and Controllers Intended for Mobile Cloud Computing. Appl. Sci. 2025, 15, 1931.

[13] Purmani, S. S. R. (2025). Streamlining IT operations and service management with agile frameworks. European Journal of Advances in Engineering and Technology, 12(4), 76–81.

[14] Korkmaz, A.; Bulut, S.; Talan, T.; Kosunalp, S.; Iliev, T. Enhancing Firewall Packet Classification through Artificial Neural Networks and Synthetic Minority Over-Sampling Technique: An Innovative Approach with Evaluative Comparison. Appl. Sci. 2024, 14, 7426.

[15] Marques, C.; Malta, S.; Magalhães, J. DNS Firewall Based on Machine Learning. Future Internet 2021, 13, 309.

[16] Madhloom, J.K.; Noori, Z.H.; Ebis, S.K.; Hassen, O.A.; Darwish, S.M. An Information Security Engineering Framework for Modeling Packet Filtering Firewall Using Neutrosophic Petri Nets. Computers 2023, 12, 202.