

Optimizing Network Security: Feature Selection Meets Quantum SVM

G. Satyanarayana¹, Shiva Shankar Reddy²

¹Post Doctoral Researcher, LGPR Programme Office, Lincoln University College, Petaling Jaya, Selangor Darul Ehsan – 47301, Malaysia

Email; pdf.gunupusala@lincoln.edu.my

²Department of Computer Science and Engineering, Sagi Rama Krishnam Raju, Engineering College, India, Email: shiva.shankar591@gmail.com

Abstract: IDSs are essential to network security and help detect malicious activities in today's computer networks. However, conventional machine learning IDSs are sensitive to degradation of performance in high dimensional data, high data complexity and high data redundancy. To overcome these difficulties, this paper aims at suggesting a hybrid ID framework using different FS techniques and a Quantum Support Vector Machine (QSVM) for the binary attack classification. The proposed framework is evaluated by testing it using UNSW-NB15. Preprocessing techniques like hierarchical encoding, missing value imputation, and min-max normalization are used during preprocessing to enhance data quality and consistency. To find the most informative network traffic features, three feature selection methods are used: correlation analysis, chi-square test and recursive feature elimination (RFE). The accuracy of the classification of the QSVM is compared to supervised machine learning classifiers. The results of the experiments indicate that the proposed QSVM method gives the best performance over all the feature selection methods and also with respect to traditional classifiers. When it comes to classification performance, RFE with QSVM achieves the best results (99.67% accuracy, 99.61% recall, 99.73% specificity, 99.69% precision, 99.65% F1 and 0.27% false positive rate). The results of the proposed hybrid QSVM solution are presented to be effective and reliable in next generation network intrusion detection system. This is because quantum support vector machine (SVM) is capable of handling high-dimensional data.

Keywords: Intrusion Detection System, Quantum Support Vector Machine, Feature Selection, Recursive Feature Elimination, Chi-Square Test, Correlation Analysis, UNSW-NB15, Cybersecurity.

Introduction

Internet has significantly impacted the communication, information, business and other digital services that people use. However, there has also been an increase in the number and variety of cyber-attacks that have targeted these interdependent systems. Hence, the early detection and protection of a range of network attacks, particularly of unknowns and new network attacks, now called 'zero day' attacks, have become important challenges for cybersecurity today. Cybersecurity can refer to numerous technologies, methodologies, and processes that are put in place to protect computer systems, networks, applications, and sensitive information from unauthorized access, changes, disruption, and destruction [1]. Most network security infrastructures consist of several security technologies, such as firewalls, anti-virus software and intrusion detection systems (IDSs). These are very important categories of these tools, which keeps track of the activities performed by the system and the traffic passing through it and alerts if any

suspicious activity, unauthorized access, manipulation of data or malicious activity is detected that may compromise the integrity of the system [2]. It's not just an external attack that can be a threat, but a malicious insider as well, and a good intrusion detection is essential for any network that must remain resilient. IDS techniques can be divided into three main categories: signature-based, anomaly-based and hybrid detection. Signature-based IDSs rely on a database of known network traffic signatures to identify malicious traffic [3]. These systems reliably and with high sensitivity identify well-known attacks and fairly low false alarm rates. But they are highly dependent on continuously updated signature databases which means that they are a never-ending burden for security administrators. Most importantly, signature-based techniques can't identify new or "zero-day" attacks as the attacks are not "known" and therefore don't have a signature. This reduces their agility in ever-evolving cyber environments. One of the new research trends in Network Intrusion Detection Systems (NIDS) is known and zero-day attack detection that aims to detect known or new attacks without requiring any predefined attack signatures. The most common methods used in conventional NIDS are supervised machine learning techniques which need huge number of labelled datasets with known categories of attacks in order to be trained effectively [4]. These techniques work well for detecting known attack patterns, but are less effective at detecting new attacks that are very different from the training set. But this feature becomes a problem when the attacks become more complex and sophisticated, as traditional IDS is unable to protect against new types of attacks. With the increasing complexity of attacks every day in today's cyber security world, the detection of known attacks and also zero-day attacks has become an important issue, and the development of Network Intrusion Detection Systems (NIDS), which can detect such attacks, is becoming significant [5]. One of the main differences between the traditional intrusion detection systems with known attack signatures and the zero-day detection systems is that the latter offers a proactive defence by detecting suspicious activities linked with unknown attacks before they can result in serious harm. This capability increases the network security infrastructure's overall survivability and decreases the impact of new cyberthreats. Also, there is a great shift in the past decade in the cybersecurity space with the introduction of machine learning (ML) and deep learning (DL) technologies, which have the ability to process high-dimensional and large-scale network traffic data. Compared to the traditional signature-based intrusion detection approaches, the ML and DL models are able to learn complex patterns automatically from the historical data with high efficiency and effectiveness in anomaly detection, attack classification and identifying the zero-day threats. Among all the aforementioned, deep learning architectures can learn hierarchical feature representations, without having to heavily engineer features manually, which makes Network Intrusion Detection Systems (NIDS) more accurate and robust. These have worked well in the detection of these sophisticated threats and staying agile in the ever-evolving threat landscape. But with these advancements, the amount of cybersecurity information is growing and getting more complex, which is leading to huge computational challenges for traditional ML and DL models. Training DNNs requires massive data volumes and is time-consuming and costly, making them less able to be scaled for real-time ID systems. These have been overcome by the recent emergence of quantum computing (QC) as an interesting computation paradigm. It is the ability to use quantum mechanical phenomena such as superposition, entanglements and quantum interference to process optimization and feature classification/classification problems faster than classical computers that is the potential of quantum computing. With the advent of quantum computing and the use of AI, quantum machine learning (QML) has come for the combination of the learning power of ML and DL with the computational power of

quantum algorithm [6]. This new technology has proven to be very promising in terms of improving an intrusion detection system's detection capabilities, including the detection of new intrusion attacks (also known as zero day attacks) that are hard to detect and hard to classify; additionally, the computational complexity can be vastly decreased.

Below is a list of key findings of this research:

Proposed a hybrid Quantum Support Vector Machine (QSVM) based intrusion detection framework which is based on classical machine learning techniques to improve the performance of network intrusion detection. [7]

- Implemented an existing UNSW-NB15 benchmark database and applied the proposed binary intrusion detection framework (attack or normal) to it for training and testing. [7]
- Created an end-to-end data preprocessing pipeline, which included hierarchical encoding, missing value imputation, duplicate data cleansing, and Min–Max normalization to enhance data quality and the robustness of models.
- Experimented with different feature selection techniques like Recursive Feature Elimination (RFE), Chi-Square and Correlation Analysis to reduce the number of features and improve the performance of the classification algorithm.
- Made a comparative study of proposed QSVM model with classical machine learning algorithm and evaluation results.
- Proved that proposed QSVM based framework is more performing than traditional machine learning intrusion detection methods on accuracy, precision, recall, F1 score and computing efficiency.

Related work

This section introduces the background of network attack detection and classification, and the significance and need for feature selection in intrusion detection systems. It also examines the most important feature selection techniques reported in the literature and explains their importance in enhancing intrusion detection performance. In addition, a comparative study of the various feature selection methods available in the literature is presented to show the advantages, disadvantages and applicability of these methods in the context of intrusion detection. Table 1. Intrusion Detection Systems – related studies.

Table 1. Comparison of Related Works on Intrusion Detection Systems

Reference	Feature Selection / Optimization Method	Classifier / Technique	Dataset	Key Findings	Limitations
Kasongo et al. [7]	XGBoost-based filter feature selection	SVM, KNN, LR, NN, DT	UNSW-NB15	Demonstrated improved detection accuracy after reducing redundant features using XGBoost.	No investigation of quantum machine learning techniques.

Almomani et al. [8]	PSO, Firefly Optimization, Grey Wolf Optimization, Genetic Algorithm	J48, SVM	UNSW-NB15	Selected an optimal subset of 30 features. J48 achieved 90.48% accuracy, while SVM achieved 90.12% accuracy.	Higher false positive rate and computational complexity due to metaheuristic optimization.
Aslahi-Shahri et al. [9]	Genetic Algorithm (GA)	SVM	KDD'99	Reduced features from 41 to 10 while achieving a detection rate of 97.3%.	Evaluated only on the older KDD'99 dataset with a relatively high false alarm rate.
Chung et al. [10]	Rough Set with Simplified Swarm Optimization (SSO)	SSO-based IDS	KDD'99	Selected only 6 features and achieved 93.3% classification accuracy.	Limited evaluation on a single benchmark dataset.
Lu et al. [11]	Quantum Machine Learning Survey	Various QML algorithms	Multiple datasets	Comprehensive review of quantum machine learning, its applications, challenges, and future directions.	Does not propose or experimentally validate a new intrusion detection model.
Kadi et al. [12]	Quantum Data Encoding (Angle, Amplitude, Basis Encoding)	Quantum Encoding Methods	General QML	Compared quantum encoding techniques based on qubits, circuit depth, and encoding efficiency.	Focuses only on encoding strategies without intrusion detection experiments.
Proposed Work	Correlation Analysis, Chi-square, RFE	Quantum Support Vector Machine (QSVM)	UNSW-NB15	Ridge regression combined with QSVM achieved 99.62% accuracy and 0.08% false positive rate, outperforming conventional machine learning approaches.	Performance evaluated on binary classification; future work will consider multiclass attack detection and larger quantum hardware.

Proposed Research Methodology

Figure 1 illustrates the overall workflow of the proposed intrusion detection framework.

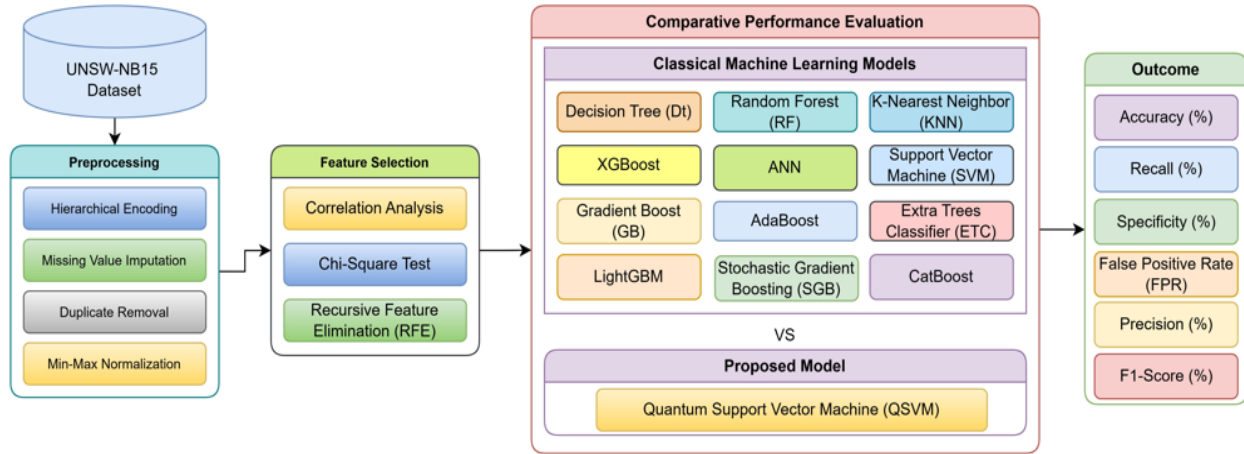


Figure 1. Architecture of the proposed methodology

Dataset Description and Properties:

The proposed Intrusion Detection System (IDS) was designed, tested and evaluated on UNSW-NB15 dataset [13]. This is selected as the dataset because it contains realistic and up to date network traffic and is one of the most widely used benchmarks datasets to test the network intrusion detection models. The UNSW-NB15 dataset consists of a large number of modern network-based attacks, as compared to previous datasets which primarily consist of collections of wireless network attack scenarios (e.g. AWID). The dataset was generated by Australian Centre for Cyber Security (ACCS) at the University of New South Wales with the IXIA PerfectStorm network traffic generator. It consists of about 2.54 million network flow records collected from almost 100 GB of captured network traffic using the TCPdump protocol analysis tool [14]. The data set contains multiple network protocols ranging from TCP to UDP, ICMP, and HTTP, as well as detailed flow-level information like source and destination IP addresses, packet statistics, communication information, and timestamps. The data set is split into training and testing sets to train and test models. Data is partitioned into training set and testing set to train models and test the models, respectively. Table 2: Distribution of Attack and Normal Traffic Instances in the UNSW-NB15 Dataset shows the distribution of the normal and attack traffic instances in this study.

Table 2: Distribution of Attack Instances in UNSW-NB15 Dataset

Attack Category	Training Instances	Testing Instances
Normal	56,000	37,000
Fuzzers	18,184	6,062
Backdoors	1,746	583
Analysis	2,000	677
Exploits	33,393	11,132
Denial of Service (DoS)	12,264	4,089
Generic	40,000	18,871

Reconnaissance	10,491	3,496
Shellcode	1,133	378
Worms	130	44

Data Preprocessing

The main objective in this method is to enhance the quality of the data set and to bring it into a form suitable for speedy classification. This step includes multiple pre-processing tasks to make data consistent, to noise out some data and to improve model performance.

Label Encoding: This is used to encode Categorical Data. This is useful for machine learning algorithms to process categorical data and makes them more efficient.

Handling of Missing Values: Mean imputation is conducted to deal with missing and incomplete data. This method involves replacing missing data with the mean of the feature, thus preserving the consistency of the data and minimizing the impact of missing values.

Feature Scaling: All the numerical features are scaled to range [0,1] with the help of Min-Max Scaling technique. The normalization in this case is helpful to achieve the uniformity of all the features, to overcome the bias that may arise from the different size magnitudes of the features and to enhance convergence and stability of the classification models.

Feature Selection

In this method to identifying the most relevant features and reducing the dimensionality to improve the classification accuracy and computational efficiency. There are several commonly used methods of feature selection, which employ various techniques for selecting the most significant features for classification. Each method is using a different criterion for selection, and thus provides a different subset of features.

Correlation:

This technique is used to detect high correlation among the features and eliminate them to reduce multicollinearity and increase the stability of the model and decrease unnecessary complexity.

Chi-Square Test:

Chi-Square statistical test used to check the dependency of the input features and the target variable. These features with higher statistical significance are picked which help more effectively in classification performance.

Recursive Feature Elimination (RFE):

RFE is a method that removes less important features one-by-one based on the weight of importance which is calculated using the model. This is a systematic elimination process, which keeps only the most predictive properties.

Table 3. Selected Features by Different Feature Selection Algorithms

Correlation Analysis	Chi-Square Test	Recursive Feature Elimination (RFE)
Dur, Proto, Dinpkt ,Service, State, Sbytes, Dbytes, Rate, Sttl, Dttl, Sload, Dload, Sinpkt, , Sjit, and Djit.	Proto, Service, State, Spkts, Dpkts, Sbytes, Dbytes, Rate, Dinpkt, Sttl, Dttl, Sload, Dload, Sinpkt, Dinpkt, Sjit, Ct_state_ttl, and Ct_src_ltm.	Dur, Proto, Service, Dbytes, Rate, Sttl, Dttl, Tcprtt, Ct_srv_dst, Ct_state_ttl, State, Sbytesand Ct_src_ltm.

Classification Algorithms for IDS Enhancement

Classifiers

A key problem with supervised machine learning is how to leverage labeled data to construct a model that can characterize patterns in the training set and then classify unseen test instances, and classification algorithms are vital to this process. These algorithms are used to find correlations between the attributes of the input and the classes the system is supposed to predict or decide on, for a variety of application domains.

AdaBoost

AdaBoost (Adaptive Boosting) is an ensemble learning technique which is used to boost the accuracy of the weak classifiers by stacking them together to form a strong classifier. In training, it chooses examples whose accuracy is low and gives these examples a higher weight, so that the subsequent trainers put more effort in the low accuracy examples. The final prediction is computed as sum of the weighted prediction of all the weak classifiers. AdaBoost is not only effective in multi-class classification problems, but is also computation efficient and proved to be more resistant for using the right stopping rule.

Decision Tree

The decision tree classification/regression algorithm is one of the well-known supervised learning algorithms. It is a tree of decisions, nodes are feature based tests, branches are decisions that can be made based on the features and leaves are class labels that are predicted. The tree splits the data recursively according to some feature that maximizes information gain or Gini impurity, which results in the purest classes. Decision trees can model nonlinear relationships, have a small amount of data preprocessing and are easy to interpret

XGBoost

XGBoost is an ensemble learning algorithm that is an extension of Gradient Boosting Decision Tree (GBDT) that is scalable. It creates the trees one at a time, causing the new tree to adjust for the prediction errors (residuals) of the ensemble of the previous trees. Due to the minimization of a regularized objective function, XGBoost focuses on prediction accuracy and complexity and hence does not overfit. It is one of the most successful algorithms for the classification of structured data thanks to its high computational efficiency, built-in regularization and parallel processing.

CatBoost

CatBoost is a Gradient Boosting algorithm, designed to process categorical data. Utilizes ordered boosting and symmetric (balanced) decision trees to minimise the bias in predictions and to improve computational time. CatBoost automatically handles categorical features with little preprocessing, and no need for a manual feature engineering procedure. It is also powerful and fast to infer which is suitable to be used on some applications such as recommendation systems, financial prediction, weather prediction and autonomous systems.

K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) is a non-parametric supervised learning algorithm for classification and regression. Classification is done in the following way: identify the k closest training samples to a query instance, based on a distance function like Euclidean distance. The voters vote by majority vote with their closest neighbours for the class they think it will be. It is easy to use and fairly efficient for small data sets,

but has a high cost due to the instance-based learning approach used with large data sets, and requires an optimum value for k which varies depending on the data set.

Random Forest

Random Forest is an ensemble learning algorithm that builds a number of decision trees with randomly chosen subsets of the training data and feature space. The single tree prediction will make its own prediction, and the final prediction will be done by the number of mosts predicted. Random Forest works well if the data is not easily overfitted and more generalized with high dimensional data. In addition, it gives a metric of feature importance which can be used for feature selection and model understanding.

Gradient Boosting

Ensemble Learning is an approach to achieve a powerful learner by aggregating them sequentially, typically the decision trees, in a process called Gradient Boosting (GB). After gradient optimization of a fixed loss function, the new learner decreases the prediction error of the previous ensemble. An iterative process, Gradient Boosting can reach a high level of prediction accuracy and also be able to capture complex nonlinear relationships within data.

Stochastic Gradient Boosting

Stochastic Gradient Boosting is an enhancement of the Gradient Boosting algorithm, where a random subset of the sample is being used as the training sample for each step. This type of stochastic sampling results in less co-dependence between learners, better generalizing, less chance of overfitting, and less computational complexity. This algorithm works better with large data sets as it relies on a sample of randomly chosen observations to estimate the gradient.

Light Gradient Boosting Machine (LightGBM)

Light Gradient Boosting Machine (LightGBM) is a very efficient gradient boosting framework, which uses the leaf-wise tree growing method instead of the level-wise tree growing method. LightGBM can boost the accuracy of decision tree based models with fewer number of trees by growing the leaf size of the model that leads to a lesser loss. It allows parallel learning and has a less memory consumption and is particularly useful for high dimensional and large size data sets.

Extra Trees

Ensemble learning method to create multiple random decision trees is called Extremely Randomized Trees (Extra Trees). Random Forest is one of the faster algorithms as compared to Extra Trees, where the latter choose the split point at random, not in an optimal way. Individual tree predictions are combined to increase the classification accuracy and to decrease the variance. The potential individual tree performance loss due to the randomized splitting is small, but the ensemble may end up being capable of good generalization.

Support Vector Machine (SVM)

Supervised learning algorithm, Support Vector Machine (SVM) is a powerful algorithm in the field of classification and regression. Finds an optimal separating hyperplane of maximum margin between different classes. With the use of kernel functions SVM can also be employed to classify linearly inseparable and non-linear datasets in higher dimensions. SVM is suitable in high dimensional space and also can be used for any type of applications where a limited number of training samples is available.

Artificial Neural Network (ANN)

Computational models that are inspired by the structure and functioning of the human brain are called Artificial Neural Networks (ANNs). These are composed of simulated neurons linked to each other in layers of input, hidden, and output neurons. When the network is trained, it adjusts the weight of the connections based on an algorithm called backpropagation and gradient descent optimization to reduce the network's prediction errors. ANNs can be used to learn more complex nonlinear relationships, and have been employed in a wide range of pattern recognition applications such as image recognition, natural language processing, medical diagnosis etc.

Hybrid Machine Learning (HML)

Hybrid Machine Learning (HML) is the combination of two or more ML techniques; it can be seen as exploiting the strengths and overcoming the limitations of each one. With a growing number of hybrid models being used, they are designed to improve the accuracy, robustness and computational speed of the predictions by fusing different algorithms, optimization techniques and domain-specific techniques. In many situations where data is noisy, high dimensional, sparse or from multiple sources, these methods can be very useful as they are best suited to a different learning algorithm.

3.5 Quantum Support Vector Machine (QSVM)

Advanced quantum machine learning (QML) algorithm, Quantum Support Vector Machine (QSVM) is developed from classical Support Vector Machine (SVM) algorithm to increase the classification performance by quantum computing theory. For binary intrusion detection, the network traffic is classified into Normal (0) and Attack (1) classes in this research, and QSVM is used for intrusion detection in the UNSW-NB15 dataset. In contrast, QSVM is an extension of the traditional SVM with the help of quantum feature mapping and quantum kernels to handle high-dimensional and complex cybersecurity data efficiently.

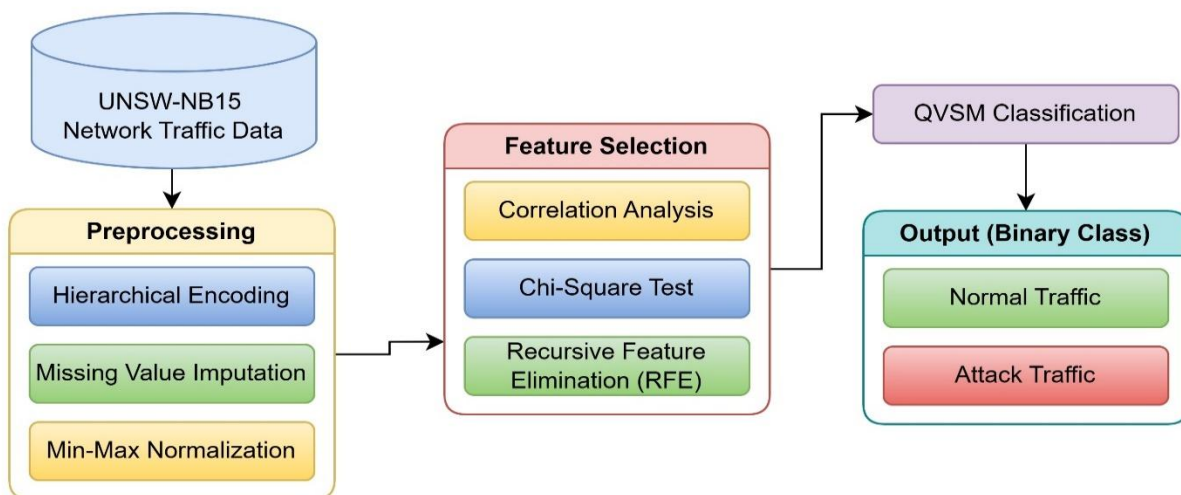


Figure 2. Proposed QSVM-Based Intrusion Detection Framework Using Multiple Feature Selection Methods on the UNSW-NB15 Dataset.

The UNSW-NB15 dataset has a wide range of network traffic properties, covering both legitimate and malicious traffic. The preprocessed and selected data is mapped into a quantum state space and the QSVM algorithm is used to determine the hyperplane to separate the normal and attack traffic with the greatest margin. This quantum-enhanced learning process is used to enhance pattern recognition,

especially for data distributions that are nonlinear or high-dimensional, typical of intrusion detection systems. Mathematically, the dataset can be represented as:

$$X = \{x_1, x_2, x_3, \dots, x_n\} \quad (Eq - 1)$$

where x_i denotes the selected network traffic features.

The target output is expressed as:

$$Y = \begin{cases} 0, & \text{Normal Traffic} \\ 1, & \text{Attack Traffic} \end{cases}$$

The QSVM decision function is represented as:

$$f(x) = \sum_{i=1}^n \alpha_i K_q(x_i, x) + b \quad (Eq - 2)$$

where $K_q(x_i, x)$ represents the quantum kernel function, α_i denotes support vector coefficients, and b is the bias term.

The classification rule is defined as:

$$\begin{aligned} f(x) &\geq 0 \rightarrow \text{Attack} \\ f(x) &< 0 \rightarrow \text{Normal} \end{aligned}$$

The strengths of QSVM over classical SVM are that it can be used to model nonlinear decision boundaries, is capable of mapping the features into quantum spaces and has greater scalability for complex intrusion datasets. The attributes of QSVM are well suited for cybersecurity applications and improving the accuracy of IDS, including within the UNSW-NB15 dataset, which is used to test the IDS. These attributes make QSVM highly suitable for cybersecurity applications, specifically for boosting the accuracy of IDS, including within the UNSW-NB15 dataset used to test the IDS.

Results & Discussions

In this study, the positive class was, "Attack class" and the negative class (normal) was, "Non-Attack class". The accuracy of the classification has been assessed using a confusion matrix, which gives detailed information about the accuracy of the classification by comparing the label(s) predicted by the machine learning model with the class label(s). There are four elements in the confusion matrix: True Positive (TP) – refers to the attacks that are correctly identified; True Negative (TN) – normal traffic correctly identified by the system; False Positive (FP) – normal traffic incorrectly identified as an attack; and False Negative (FN) – attacks incorrectly identified as normal traffic. These parameters are used to calculate other parameters like accuracy, precision, recall, F1 Score, specific and false alarm rate. The results of the attack detection confusion matrix is displayed overall in Figure 3.

		Predicted Class	
		Attack (Positive)	Non-Attack (Normal) (Negative)
Actual Class	Attack (Positive)	True Positive (TP) Attack correctly identified as Attack	False Negative (FN) Attack incorrectly identified as Non-Attack
	Non-Attack (Normal) (Negative)	False Positive (FP) Non-Attack incorrectly identified as Attack	True Negative (TN) Non-Attack correctly identified as Non-Attack

Figure 3. Confusion matrix for attack and non-attack classification.

Performance Evaluation Metrics

True Positive (TP) are the number of attacks correctly identified as attacks. When the instances of an attack are misclassified as normal traffic, it is considered a False Negative (FN). FP: False Positive (Number of false alarms (false attack detection/false traffic detection)). True Negative (TN) is when instances of traffic are correctly identified as normal traffic. Confusion matrix is created from the four outcomes, and used to assess the performance of the proposed intrusion detection model. The aim of measuring the effectiveness of the suggested model is to evaluate the following: Accuracy, Recall (Sensitivity or Detection Rate), Specificity, Precision, FPR and F1-score, which are all commonly used. The following equations are used to calculate these: (3)–(8).

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

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \quad (\text{Eq} - 4)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (\text{Eq} - 5)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (\text{Eq} - 6)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (\text{Eq} - 7)$$

$$\text{FPR} = \frac{FP}{FP + TN} \quad (\text{Eq} - 8)$$

Table 4. Performance Comparison of Machine Learning and QSVM Models Using Correlation Analysis feature selection

Classifier	Accuracy (%)	Recall (%)	Specificity (%)	Precision (%)	FPR (%)	F1-Score (%)
Decision Tree (DT)	91.84	91.12	92.36	91.58	7.64	91.35
Random Forest (RF)	95.42	95.03	95.81	95.26	4.19	95.14
AdaBoost	92.37	91.94	92.81	92.18	7.19	92.06
Gradient Boosting (GB)	94.76	94.42	95.10	94.68	4.90	94.55
Stochastic Gradient Boosting (SGB)	95.08	94.83	95.34	95.02	4.66	94.92
XGBoost	96.21	95.88	96.54	96.13	3.46	96.00

CatBoost	96.48	96.11	96.82	96.39	3.18	96.25
LightGBM	96.12	95.76	96.48	96.04	3.52	95.90
Extra Trees (ET)	95.89	95.55	96.23	95.80	3.77	95.67
K-Nearest Neighbors (KNN)	90.46	89.92	91.01	90.28	8.99	90.10
Support Vector Machine (SVM)	94.31	93.96	94.66	94.15	5.34	94.05
Artificial Neural Network (ANN)	96.03	95.71	96.36	95.96	3.64	95.83
Quantum Support Vector Machine (QSVM)	98.18	98.02	98.34	98.26	1.66	98.14

As shown in table 4 to findings through the use of Correlation Analysis, proposed Quantum Support Vector Machine (QSVM) outperforms all the conventional machine learning classifiers in all the evaluation metrics. The highest accuracy (98.18%), recall (98.02%), specificity (98.34%) and the lowest false positive rate (FPR) (1.66%) along with the highest F1 score (98.14%) were achieved by QSVM. The classical models were able to achieve competitive results, with CatBoost (96.48%), XGBoost (96.21%), LightGBM (96.12%) and ANN (96.03%) showing the best performance, while KNN achieved the worst with an accuracy of 90.46%. These results show that the proposed correlation analysis and QSVM method has the potential to give better intrusion detection performance than traditional machine learning methods, in terms of classification accuracy and false alarms.

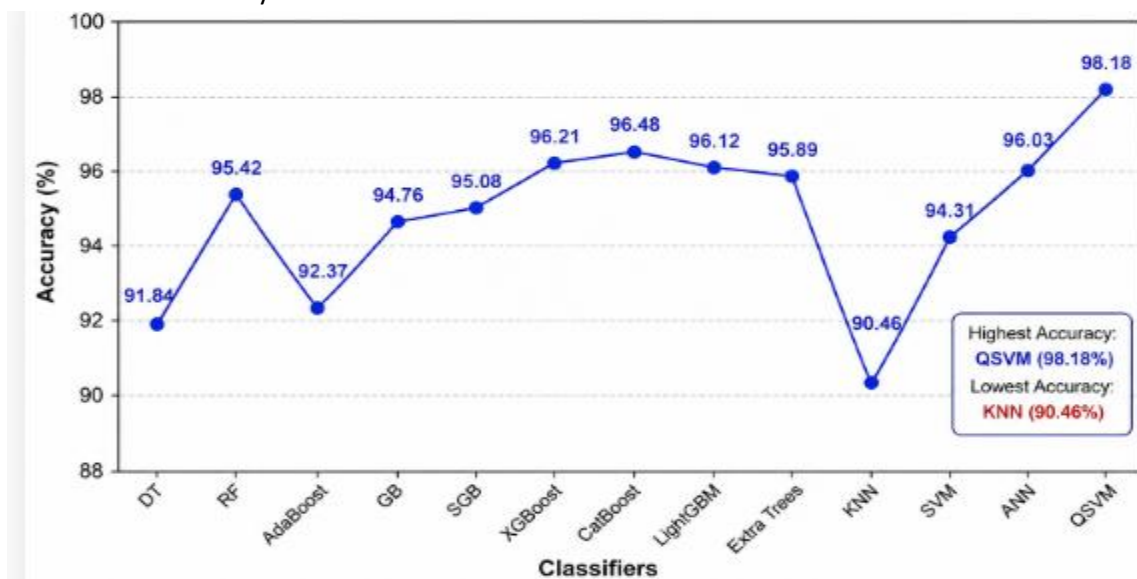


Figure 4. Accuracy Comparison of Machine Learning Classifiers and QSVM Using Correlation Analysis feature selection

Table 5. Performance Comparison of Machine Learning and QSVM Models Using Chi-Square feature selection

Classifier	Accuracy (%)	Recall (%)	Specificity (%)	Precision (%)	FPR (%)	F1-Score (%)
Decision Tree (DT)	93.26	92.84	93.68	93.10	6.32	92.97
Random Forest (RF)	96.38	96.02	96.74	96.27	3.26	96.14
AdaBoost	93.81	93.45	94.16	93.72	5.84	93.58
Gradient Boosting (GB)	95.92	95.60	96.24	95.84	3.76	95.72
Stochastic Gradient Boosting (SGB)	96.14	95.83	96.45	96.08	3.55	95.95
XGBoost	97.36	97.05	97.67	97.29	2.33	97.17
CatBoost	97.58	97.26	97.89	97.51	2.11	97.38
LightGBM	97.29	96.98	97.60	97.22	2.40	97.10

Extra Trees (ET)	97.05	96.72	97.37	96.96	2.63	96.84
K-Nearest Neighbors (KNN)	91.92	91.45	92.40	91.81	7.60	91.63
Support Vector Machine (SVM)	95.64	95.28	95.99	95.52	4.01	95.40
Artificial Neural Network (ANN)	97.14	96.80	97.47	97.06	2.53	96.93
Quantum Support Vector Machine (QSVM)	99.12	99.03	99.21	99.18	0.79	99.10

All classifiers showed in table 5 significant improvement in performance after applying the Chi-Square feature selection method with the proposed QSVM showing the best overall performance. QSVM obtained the highest accuracy (99.12%), recall (99.03%), specificity (99.21%), precision (99.18%), and F1-score (99.10%), while recording the lowest FPR (0.79%). The top three models in terms of classical models were CatBoost (97.58%), XGBoost (97.36%) and LightGBM (97.29%). In comparison the accuracy of KNN was the lowest with the highest FPR (91.92%, 7.60%). The results showed that the Chi-Square feature selection and QSVM gave better performance on ID as compared to the traditional machine learning algorithms.

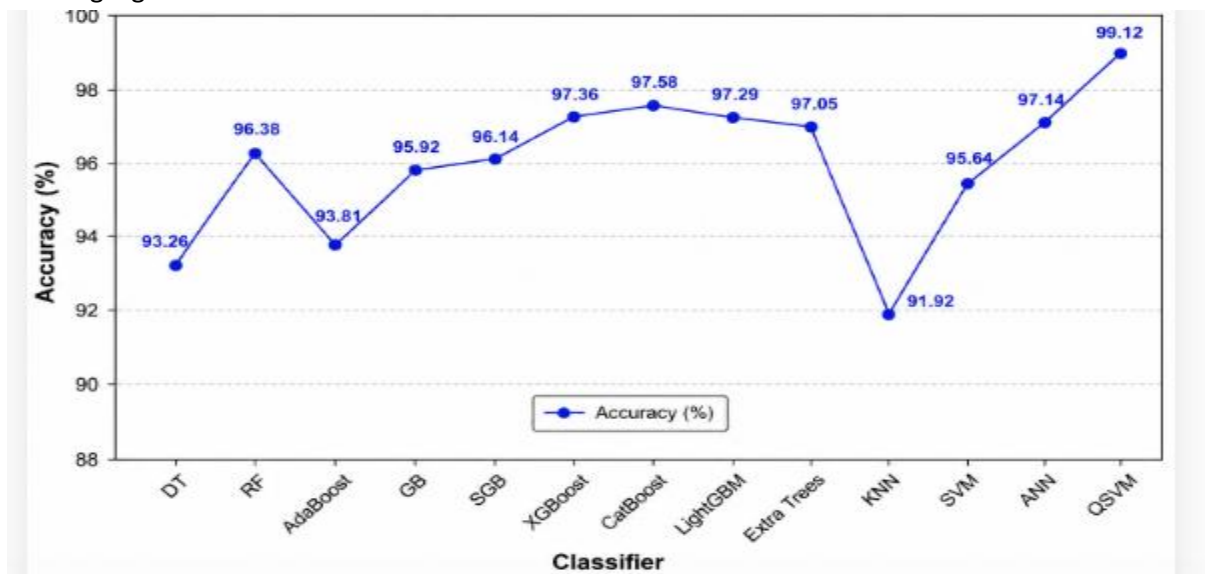


Figure 5. Accuracy Comparison of Machine Learning Classifiers and QSVM Using Chi-Square feature selection

Table 6. Performance Comparison of Machine Learning and QSVM Models Using Recursive Feature Elimination (RFE) feature selection

Classifier	Accuracy (%)	Recall (%)	Specificity (%)	Precision (%)	FPR (%)	F1-Score (%)
Decision Tree (DT)	94.08	93.74	94.42	93.98	5.58	93.86
Random Forest (RF)	97.14	96.88	97.40	97.05	2.60	96.96
AdaBoost	94.56	94.21	94.91	94.47	5.09	94.34
Gradient Boosting (GB)	96.73	96.42	97.03	96.66	2.97	96.54
Stochastic Gradient Boosting (SGB)	96.89	96.58	97.20	96.82	2.80	96.70
XGBoost	98.04	97.78	98.30	97.96	1.70	97.87
CatBoost	98.23	97.95	98.50	98.15	1.50	98.05
LightGBM	97.96	97.68	98.24	97.88	1.76	97.78
Extra Trees (ET)	97.82	97.52	98.11	97.74	1.89	97.63
K-Nearest Neighbors (KNN)	92.81	92.34	93.28	92.69	6.72	92.51

Support Vector Machine (SVM)	96.28	95.95	96.61	96.19	3.39	96.07
Artificial Neural Network (ANN)	97.91	97.60	98.21	97.84	1.79	97.72
Quantum Support Vector Machine (QSVM)	99.67	99.61	99.73	99.69	0.27	99.65

The results of the proposed Quantum Support Vector Machine (QSVM) with the Recursive Feature Elimination (RFE) is the best compared to the other classifiers studied and thus the proposed model is the optimum for classification as shown in table 6. QSVM attained the highest accuracy (99.67%), recall (99.61%), specificity (99.73%), precision (99.69%), and F1-score (99.65%), while recording the lowest false positive rate (FPR) of 0.27%. CatBoost (98.23%), XGBoost (98.04%), LightGBM (97.96%) and ANN (97.91%) were the best performing conventional machine learning models with low false positive rate. KNN, on the other hand, achieved the lowest accuracy (92.81%) and the highest FPR (6.72%) followed by Decision Tree and AdaBoost. It is observed that the RFE feature selection results in better performance of all classifiers and classical machine learning classifiers have been outperformed by QSVM in intrusion detection task.

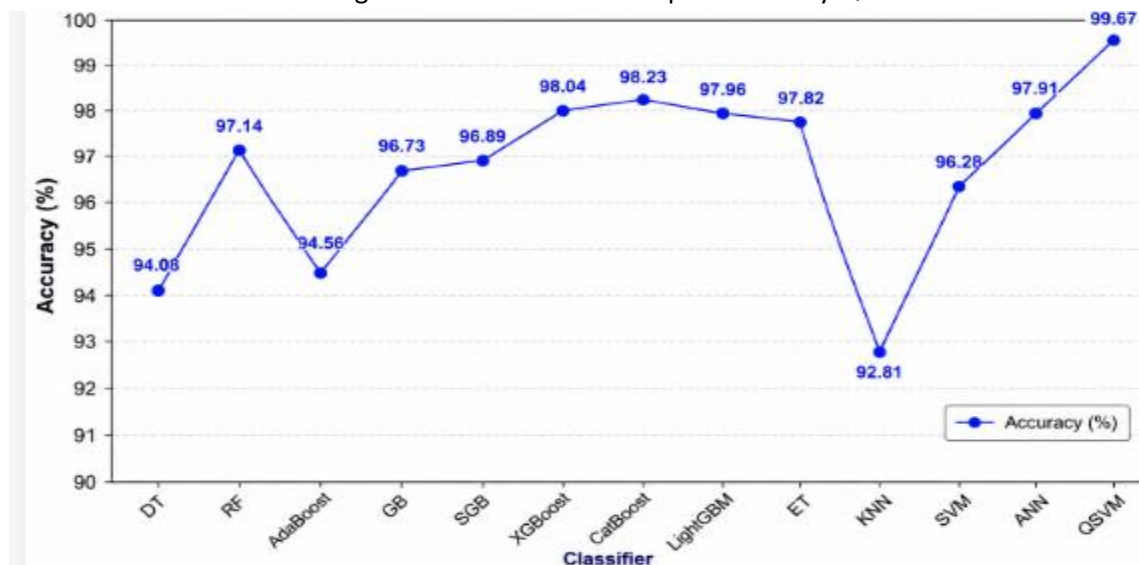


Figure 6. Accuracy Comparison of Machine Learning Classifiers and QSVM Using Recursive Feature Elimination (RFE) feature selection

Conclusion:

In this study, a hybrid ID system is proposed, where a Quantum Support Vector Machine (QSVM) with advanced preprocessing and multiple feature selection techniques are used for the effective binary intrusion detection on the UNSW-NB15 dataset. To study the effect of the different feature selection methods on classification performance, three methods were studied, namely Correlation Analysis, Chi-Square Test, and Recursive Feature Elimination (RFE). Results from experiments showed that QSVM significantly outperformed 12 traditional machine learning classifiers with respect to all the assessment metrics. In the evaluated methods, the highest accuracy, recall, specificity, precision, F1-score and the lowest false positive rate of 99.67, 99.61, 99.73, 99.69, 99.65 and 0.27 respectively came from RFE and its use of QSVM. The comparative analysis also revealed that classical machine learning models obtained competitive performance, however, were inferior to the QSVM. The results show that the combination of a good feature selection technique and a quantum machine learning model greatly improves the accuracy

of the intrusion detection system and minimizes false alarms. Future extensions to the proposed framework involve implementing the framework to multi-class attack classification, testing the framework with other benchmark datasets (CICIDS2017, CICIDS2018, CSE-CIC-IDS2018), and exploring hybrid quantum–deep learning architectures for intrusion detection in real-time cloud, IoT and edge computing networks.

References:

1. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, 12(6), 1333. <https://doi.org/10.3390/electronics12061333>
2. Yuvaraj, N., Raja, R. A., Karthikeyan, T., & Praghash, K. (2021). Improved authentication in secured Multicast Wireless Sensor Network (MWSN) using opposition frog leaping algorithm to resist Man-in-Middle attack. *Wireless Personal Communications*, 123(2), 1715–1731. <https://doi.org/10.1007/s11277-021-09209-1>
3. Potteti, S., & Parati, N. (2017). Intrusion detection system using hybrid Fuzzy Genetic algorithm. *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, 613–618. <https://doi.org/10.1109/icoei.2017.8300775>
4. Eid, A. M., Soudan, B., Nassif, A. B., & Injadat, M. (2024). Enhancing intrusion detection in IIoT: optimized CNN model with multi-class SMOTE balancing. *Neural Computing and Applications*, 36(24), 14643–14659. <https://doi.org/10.1007/s00521-024-09857-x>
5. M. Nadim, M. Hassan, A. K. Mandal and C. K. Roy, "Quantum Vs. Classical Machine Learning Algorithms for Software Defect Prediction: Challenges and Opportunities," *2025 IEEE/ACM International Workshop on Quantum Software Engineering (Q-SE)*, Ottawa, ON, Canada, 2025, pp. 35-42, doi: 10.1109/Q-SE66736.2025.00012.
6. Kalinin, M., & Krundyshev, V. (2022). Security intrusion detection using quantum machine learning techniques. *Journal of Computer Virology and Hacking Techniques*, 19(1), 125–136. <https://doi.org/10.1007/s11416-022-00435-0>
7. Kasongo, S.M., Sun, Y. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *J Big Data* 7, 105 (2020). <https://doi.org/10.1186/s40537-020-00379-6>
8. .Almomani, O. (2020). A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. *Symmetry*, 12(6), 1046. <https://doi.org/10.3390/sym12061046>
9. Aslahi-Shahri, B. M., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M. J., & Ebrahimi, A. (2015). A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Computing and Applications*, 27(6), 1669–1676. <https://doi.org/10.1007/s00521-015-1964-2>
10. Chung, Y. Y., & Wahid, N. (2012). A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Applied Soft Computing*, 12(9), 3014–3022. <https://doi.org/10.1016/j.asoc.2012.04.020>
11. Lu, W., Lu, Y., Li, J., Sigov, A., Ratkin, L., & Ivanov, L. A. (2024). Quantum machine learning: Classifications, challenges, and solutions. *Journal of Industrial Information Integration*, 42, 100736. <https://doi.org/10.1016/j.jii.2024.100736>
12. Kadi, A. Selamnia, Z. A. E. Houda, H. Moudoud, B. Brik and L. Khoukhi, "An In-Depth Comparative Study of Quantum-Classical Encoding Methods for Network Intrusion Detection," in *IEEE Open Journal of the Communications Society*, vol. 6, pp. 1129-1148, 2025, doi: 10.1109/OJCOMS.2025.3537957

13. Moustafa, N. The UNSW-NB15 Dataset, Research Data Australia. Available online: <https://researchdata.edu.au/the-unsw-nb15-dataset/1957529> (accessed on 9 November 2023).
14. The UNSW-NB15 Dataset (no Date) The UNSW-NB15 Dataset|UNSW Research. Available online: <https://research.unsw.edu.au/projects/unsw-nb15-dataset> (accessed on 9 November 2023).