scientific reports

Check for updates

OPEN

Intrusion detection system based on machine learning using least square support vector machine

Pratik Waghmode, Manideep Kanumuri, Hosam El-Ocla✉ & Tanner Boyle

Security solutions in the cyber world are essential for enforcing protection against network vulnerabilities and data exploitation. Unauthorized access or attack can be avoided in critical systems using a comprehensive approach via an effective intrusion detection system (IDS). Traditional intrusion detection techniques are no longer accurate and effective enough to handle the demands of the big data age. Machine learning (ML) methods can be utilized for intrusion detection since the classifier's performance has significantly increased over the past decade. A significant limitation of most ML-based IDSs is that they often generate alerts for false predictions. This is owing to misclassifications that tend to occur more frequently than actual threats. Despite their effectiveness, these conventional ML-based IDSs often face difficulties scaling to meet the demands of big data. The increasing volume and complexity of datasets pose various challenges, such as high dimensionality, multiple data sources, and the need for a dependable infrastructure. Consequently, the accuracy of an ML model likely declines when irrelevant features are included from a vast dataset. In this paper, the exhaustive feature selection algorithm is employed to assess every possible combination of features in a dataset to evaluate its performance. The selection is based on identifying the feature subset with the highest accuracy. Hence, an ML-based complete security solution is introduced for network intrusion detection using the supervised framework. This framework utilizes quantum-inspired least square support vector machine (LS-SVM) classifier. This algorithm is used to enhance the classification accuracy in terms of reducing false predictions while minimizing the training time. The hyperparameters of our model are tuned by utilizing those selected features to maximize the accuracy. The model developed is verified using three different datasets, which have been widely applied to intrusion detection. The model achieves high detection performance, with accuracy values of 99.3% for NSL-KDD, 99.5% for CIC-IDS-2017, and 93.3% for UNSW-NB15. Precision remains at 1.00 for CIC-IDS-2017 and UNSW-NB15, while recall reaches 1.00 for CIC-IDS-2017, 0.99 for NSL-KDD, and 0.98 for UNSW-NB15. F1-scores follow the same trend, reflecting the classifier's robust prediction capabilities. In addition, our model demonstrates competitive testing time efficiency in 2.8 s for NSL-KDD, 1.0s for CIC-IDS-2017, and 2.8s for UNSW-NB15. Also, our model requires the minimum training time for all datasets compared to other models. These results highlight the LS-SVM-based model's suitability for real-time intrusion detection applications.

Keywords Intrusion detection system, Machine learning, Deep learning, Feature selection, Cyber security, Exhaustive feature selection, Support vector machine, Least square support vector machine

Cyber security faces tremendous hurdles as a result of the rapid development and widespread adoption of technologies like 5G, IoT, cloud computing, and others that have increased network scale, real-time traffic, and cyber attack complexity and diversity. The network Intrusion Detection System (IDS) serves as the second line of defense behind the firewall and is responsible for accurately identifying hostile network attacks, providing real-time monitoring and dynamic security measures. A network attack is an effort to access a company's network without authorization with the intent of stealing information or carrying out other destructive behavior[1]. As a result, a network system needs to include several security tools to shield sensitive information and services from danger. IDS is a solution that can be implemented through software or hardware and is used to detect malicious act or violation schemes in networks. It keeps track of any malicious activity or security lapses occurring on the network and alerts the administrator to any possible hazards. Cyber attacks are getting more dangerous every

Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada. ✉email: hosam@lakeheadu.ca

day. Modern networked corporate environments require a high level of security for safe and reliable information sharing across businesses to stay up with the ever-evolving cyber threats[2].

Machine learning (ML), a branch of artificial intelligence, has been widely employed in recent years[3,4] to enhance intrusion detection by enabling sophisticated automation and prediction. Modifiable, replicable, and extensible datasets are handled using ML approaches. These methods teach IDSs how to recognize and combat both seen and unseen threats, enabling IDSs to become more resilient. The feature selection method used in this work is the Exhaustive Feature Selection. Feature selection[5] is the process of choosing the most reliable, non-redundant, and pertinent features to be exploited in a classification model. As the quantity and variety of data sets increase, it is crucial to gradually reduce their size. Feature selection's primary objectives are to enhance the predictive model's performance and lower the modeling's computational expense[6]. It is essential to increase accuracy and cut down on model training time. Deep learning (DL) has made incredible progress in the disciplines of computer vision (CV), autonomous driving (AD), and natural language processing (NLP) in recent years, thanks to its potent automatic feature extraction capability[7]. DL is widely used in intrusion detection for traffic classification, which has emerged as a hot topic in modern research. To classify network traffic anomaly detection as a problem, DL uses a training model to extract probable characteristics from high-dimensional data[8].

Using knowledge about previous network attacks that have been perpetrated, IDS can evaluate information on the node access and determine whether the traffic is potentially harmful. IDS does signature analysis, just like many antivirus programs, to find out whether to alert the administrator. All security measures based on signature analysis are vulnerable to attack, nevertheless. It is important to consider and promptly identify behaviors aimed at getting around the IDS and violating its rules to increase the efficacy of how it operates. Various techniques can be used by an attacker to avoid detection by the system.A major drawback of an IDS is that it frequently warns of false positives (erroneous alerts). False positives are frequently more common than genuine threats. Particularly, engineers likely have to spend time responding to false positives even if an IDS is adjusted to limit their frequency. Real attacks may go undetected or be overlooked if they don't take care to watch out for the false positives[9] and this is what inspired us to pursue a study to propose an efficient IDS. As a result, we aim at proposing an ML model that maximizes the accuracy through utilizing a feature selection and classifier methods that would ultimately reduce obviously the false predictions, particularly positives. of their efficiency, traditional ML-based IDSs encounter challenges in scaling to accommodate the needs of big data of datasets generated as a result of the large volumes of network traffic data. Accordingly, the performance of an ML model in terms of accuracy and speed may degrade when irrelevant features are included in a large dataset. Imbalanced datasets pose another difficulty for predictive modeling particularly with big datasets. Making a dataset balanced facilitates the training of a model since it reduces the likelihood of the model showing bias towards a particular class.

In this paper, we develop a binary classification system to detect intrusion from three well-known datasets: NSL-KDD[10,11], CICIDS2017[12], and UNSW-NB15[13]. Our model implements the Exhaustive Feature Extraction technique combined with ADABOOST, SDG, KNN, and XGBOOST classifiers. This feature selection algorithm is utilized to examine every possible combination of features within a dataset to evaluate its effectiveness. The selection process aims to identify the subset of features that delivers the best accuracy. Therefore, a rigorous security solution based on ML is proposed for network intrusion detection through a supervised framework. With this exhaustive feature selection technique, the best features are selected for classification based on the accuracy of the classifiers used in the feature extraction, as well as the best features from each classifier. Experimental results indicate that this architecture demonstrates high detection performance in terms of superior accuracy, precision, recall, and F1-score, showcasing its significant improvement over existing methodologies. The LS-SVM classifier in our system achieves an accuracy of 99.3% on NSL-KDD, 99.5% on CIC-IDS-2017, and 93.3% on UNSW-NB15. Additionally, precision scores remain at 1.00 for CIC-IDS-2017 and UNSW-NB15, while recall values reach 1.00 for CIC-IDS-2017, 0.99 for NSL-KDD, and 0.98 for UNSW-NB15, leading to similarly high F1-scores. Furthermore, the LS-SVM-based model achieves efficient testing time performance, with 2.8 s for NSL-KDD, 1.0 s for CIC-IDS-2017, and 2.8 s for UNSW-NB15. Also, our model requires the minimum training time for all datasets compared to other models. This combination of high accuracy and computational efficiency positions our IDS as a robust solution for real-time threat detection.

Literature review

In this section, we discuss the previous work done on the intrusion detection systems. We go through various techniques and technologies proposed lately.

IDS often encounters considerable difficulties in identifying minority class attacks in imbalanced network traffic. As a result, feature selection plays a primary role in the ML model particularly when employed in IDS where cyber-attacks are real threats in data networks[16,17]. Authors of Ref.[18] proposed an ideal foundation for an image-based network IDS. The system combined an image improvement and transformation methodology with an improved feature selection flow. To attain overall efficiency, the suggested framework initially decreased the amount of characteristics. The non-image data is then converted to images. The improved photos have been utilized to effectively detect anomalies using a deep-learning classifier. In image processing, a larger image equates to greater precision. The suggested approach, however, used a combination of DeepInsight with the Gabor filter[19] to provide highly representative data by reducing the number of features. This, as a result, degraded the accuracy of the model.

In Ref.[20], authors used the ensemble feature selection together with various classification methods such as decision tree, Support Vector Machine (SVM)[21], Logistic Regression, and many more. The ensemble feature extraction method includes techniques like ANOVA, Chi-squared, LASSO, Logistic Regression with L1 Penalty,

Random Forest, Recursive Feature Elimination, Pearson Correlation, Mutual Information, and SFPR. These methods are time-consuming which affects the algorithm's complexity.

In Ref.[22], authors also used the ensemble mechanism with Bagged Trees. The bagging algorithm worked on first creating several sample sets using Bootstrap Sampling on the initial training set. Training with these sample sets using several base classifiers was considered before combining them into an integrated classifier's basic strategy. The ensemble feature extraction method, though, slowed down the productivity by taking more time to execute.

In Ref.[23], authors proposed a Difficult Set Sampling Technique (DSSTE) algorithm that reduced the majority samples and augmented the minority samples in the difficult set. This could tackle the class imbalance problem in intrusion detection so that the classifier learns the differences better in the training process. However, it was hard for the classifier to learn the preprocessed features and this restricted the use of the autonomous feature extraction capabilities.

A two-layer Improved Siam-IDS (I-SiamIDS) strategy was suggested in Ref.[24] to address the issue of class imbalance. Without utilizing any data level balancing techniques, I-SiamIDS could specify both minority and majority classes. First Siam-IDS's layer filters input samples using a binary ensemble of Siamese Neural Network, eXtreme Gradient Boosting[25], and Deep Neural Network (DNN). Attacks are then transferred to the second layer, where they are classified using the multi-class extreme Gradient Boosting classifier into several attack classes (m-XGBoost). I-SiamIDS showed a modest improvement in recall, F1 score, and precision values for the CIDDS-001[26] and NSL-KDD datasets with DoS cyber attacks.

Network-based IDSs are notorious for producing a high number of false positives and negatives. The majority of the earliest network-based IDSs employed signature-based detection to identify well-known straightforward assaults. Innovative technologies have increased the types of attacks that can be detected and achieved great accuracy by combining detection approaches[27–29]. False positive and negative rates are thereby decreased. However, it frequently needed a lot of fine-tuning and customization to accommodate the features of the observed environment[30].

AdaBoost and artificial bee colony (ABC) algorithms were used in a hybrid network-based IDS system proposed in Ref.[31] to detect anomalies. The ABC algorithm was used to choose the features and hence they were assessed and categorized using the AdaBoost algorithm. Detection efficiency is limited with several datasets.

Some rules indicate the typical behavior of users, hosts, network connections, and applications in an IDS that used an anomaly-based detection method[32]. These guidelines were created by closely observing the traits of typical activities throughout time. For instance, the average utilization time of web activity during business hours constitutes the rule for a network. Through the characteristics of the current activity, IDS could detect noticeably higher-than-expected utilization of web activity and accordingly trigger alerts. There are numerous behavioural characteristics used to produce rules including the quantity of emails sent by a user, the quantity of failed login attempts, and the quantity of packets exchanged in a specific amount of time. The ability of anomaly-based detection systems to identify previously unidentified attack types is a key benefit. Consider a scenario where a machine has a recent malware infection. The malware can use up the computer's processing power, send numerous emails, open numerous network connections, and participate in other activities that may be very different from the profiles set up for the machine[33]. An approach was proposed in Ref.[34] where a 96.9% attack detection rate was achieved with the NSL-KDD dataset. If a change in the activity happens slowly enough, IDS can classify malicious behavior as being a usual behavior and add it to its profile. With anomaly-based IDS systems, harmful activities are frequently unintentionally included as part of the rule. Furthermore, it can be challenging to write the rules correctly and this is one of the issues that occurs occasionally with anomaly-based IDSs[35]. For instance, if an event that executes significant file transfers only happens once a month, this behavior can be viewed as abnormal because it is not frequently detected, leading to the triggering of an alert. In Ref.[36], authors addressed the problem of high dimensionality in IDS and implemented five classification algorithms. Two projection approaches were utilized: PCA and random projection on NSL- KDD dataset. Detection efficiency is low in terms of accuracy. In Ref.[37], a proactive model to forecast the likelihood of an attack occurring in the data packets. In particular, the traditional IDS dataset is transformed into a time series structure and employed predictive models to anticipate upcoming malicious packets. Experiments show low performance for intrusion detection.

Problem discussion

Security is a primary objective of the IDS and it requires a mechanism to keep an eye on computer networks and systems and it employs this knowledge to recognize external threats, system exploitation, or internal attacks[38,39]. IDS is regarded as one of the fundamental security items that should be implemented in business systems. IDS can be paired with other safety measures and layered security architecture products. For instance, some systems combine an IDS with a firewall and antivirus programs. This is how IDS can be used to identify attacks that other security tools are unable to detect. IDS can identify attacks using a variety of approaches and procedures as discussed above. In this regard, studies on anomaly identification from system calls have been conducted. Even though a lot of work has been done in this field to create global databases; however, there are still gaps in datasets that should presumably be complete. In addition, anomaly-based methods can identify both unknown and recognized attacks. Nevertheless, these methods can recognize typical assaults to a certain extent. In summary, existing IDSs cannot cope with the dynamic and continuous nature of the currently developing attack types[32–41]. Data used for model training determines the performance of the ML model. Most of the current models of the IDS are not able to classify cyber-attacks accurately. Consequently, the main challenge is to improve the accuracy of the attack detection on a comparative basis.

In this paper, we present our ML-based model that uses a Quantum-inspired Least Square Support Vector Machine with exhaustive feature selection for the classification of attacks. The objective of this work is to

implement feature selection and classification modeling techniques that can classify cyber attacks accurately and efficiently. We performed data preprocessing to clean the datasets with class balancing for further analysis. A comparison of the performance analysis among various models should be conducted to determine which model is significantly better and more efficient. We compared the performance of our model with previously implemented models using validation metrics like accuracy, F1 score, recall, and precision.

Methodology

Figure 1 shows a flowchart for the methodology of our work. The components of the flowchart are explained below:

Datasets

NSL-KDD

To address some of the underlying issues with the KDD CUP 99 dataset presented in Ref.[42], the NSL-KDD dataset has been proposed in Ref.[11]. Due to the dearth of publicly available datasets for network-based IDSs, we believe this updated version of the KDD dataset can still be used as an effective benchmark to aid in comparing various intrusion detection techniques. Additionally, the NSL-KDD train and test sets have a respectable number of records. Due to this benefit, all of the data can be used for the tests instead of just a tiny sample that must

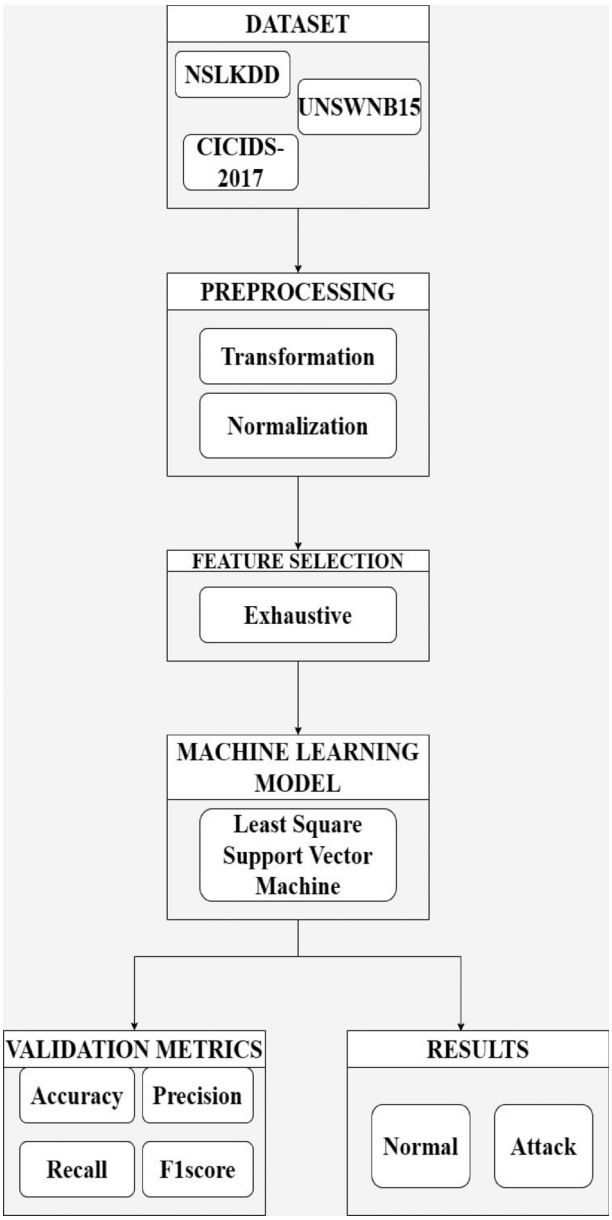


Fig. 1. Methodology of the model.

be chosen at random. As a result, evaluation findings from various research projects will be comparable and consistent.

CIC-IDS-2017

The CICIDS2017 dataset[12], which closely reflects actual real-world data, contains the most recent and benign prevalent packet capture (PCAP) attacks. It also contains the outcomes of a network traffic analysis performed using CICFlowMeter with flows categorized according to the time stamp, source and destination IP addresses, source and destination ports, protocols, and attack (CSV files). The definition of extracted characteristics is additionally available in Ref.[43].

UNSW-NB 15

The IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra synthesized the raw network packets for the UNSW-NB 15 dataset[13] to provide a hybrid of real contemporary normal activities and synthetic current attack behaviors. 100 GB of the raw traffic was captured using the tcpdump utility (e.g., Pcap files). Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are among the nine attack categories in this dataset. To produce a total of 49 characteristics with the class label, the Argus and Bro-IDS tools were utilized, and 12 methods were built. The UNSW-NB15 features.csv file contains a description of these features[44]. This dataset combines network traffic assault actions that are now synthesized with real-world modern norms. The UNSWNB15 dataset's features are produced using both established and cutting-edge techniques[45].

Data preprocessing

As part of the preparation of datasets for feeding machine learning classifiers, it is necessary to clean, sanitize, transform, normalize, and reduce the features. The data contains noise, duplicate values, missing values, and infinity values. During the extraction or input of the data, there were errors. The first step is to preprocess the data. Each sample should contain only one valid value; duplicate values should be excluded. We delete the missing values (Not a Number, Nan) and infinite values (Inf) from the data. To ensure that the data is clean and sanitized, we removed any rows whose values are ', or 'inf'. The UNSW-NB15 dataset contains a column 'ID' and it was excluded. The CICIDS2017 dataset includes a single space before each of the feature names and it has been removed.

The data encoding technique was used to convert the three datasets into numeric data. Using the min–max scaling approach, data is normalized to a range of values from zero to one. The NSL-KDD and the UNSWNB15 datasets have been prepared with almost the same number of normal and attack data instances. We have mapped 0 and 1 to normal and attack, respectively, and then data with the attack was combined into test_attack. We collect the numeric features. To make the dataset more balanced, the same number of attacks and normal samples were considered in training and testing sets similarly as in Ref.[23]. This has been fulfilled through using an undersampling mechanism. This can be accepted in the IDS as the two classes can be easily differentiated since there is minimal overlap between them. This allows the model to require less data to learn for making predictions as it is improbable for it to mix up one class with another.

After the data collection process, a total of 148,515 rows, 257,673 rows, and 2,870,343 rows of data have been incorporated into NSL-KDD, UNSW-NB15, and CIC-IDS, respectively. The datasets contain information regarding various types of intrusion attacks, such as DOS attacks, probe attacks, privilege attacks, etc.

Feature selection

Features to be included for the classification task are often challenging as it can be difficult to determine which are most valuable before evaluating. Nonetheless, conventional ML algorithms struggle to scale effectively to address the requirements of Big Data. The expanding volume and complexity of datasets present a variety of challenges, including high dimensionality, diverse data sources, and the demand for a reliable infrastructure[46]. Accordingly, the accuracy of the ML model would degrade when irrelevant features are selected out of an immense dataset. In this study, the Exhaustive Feature Selection algorithm is utilized to evaluate each possible combination of features within a dataset to determine its performance. Selection is made based on the feature subset that yields the best accuracy.

Our method for filtering features has been developed using a variety of classification methods to avoid the extraction of irrelevant features and ensure that we collect the right features. These methods include Adaboost, KNN, XGBOOST, and SGD classifiers. We have tuned the hyperparameters of our model by using the features extracted from the exhaustive feature selection method. Other key hyperparameters include learning rate (0.001), batch size (64), number of iterations (100), zero bias, activation function (ReLU), and weights that are randomly initialized. From among all the classifiers that have been used in the exhaustive feature selection, we have selected those features and a subset of features with the highest accuracy. Out of all the subsets we selected, five were fed into the classification algorithm.

Figure 2 and Table 1 show the features selected from the NSL-KDD dataset based on considering the accuracy of different classifiers used in the feature selection process such as Adaboost [3,9,47,59], KNN [1,2,47,49], SDG [3,9,15,59], and XGBOOST [3,4,47,59]. Out of these, a feature subset [3,9,47,59] has been selected and this helps in the classification process.

Figure 3 and Table 2 show the features selected from the CIC-IDS dataset based on considering the accuracy of different classifiers used in the feature selection process such as Adaboost [2,11,13,18], KNN [2,11,13,23], SDG [2,11,15,26], and XGBOOST [2,11,19,23] out of these we have selected the feature subset [2,11,13,23] which helps in the classification process.

Figure 4 and Table 3 show the features selected from the CIC-IDS dataset based on considering the accuracy from different classifiers used in the feature selection process such as Adaboost [2,3,15,24], KNN [2,11,18,24],

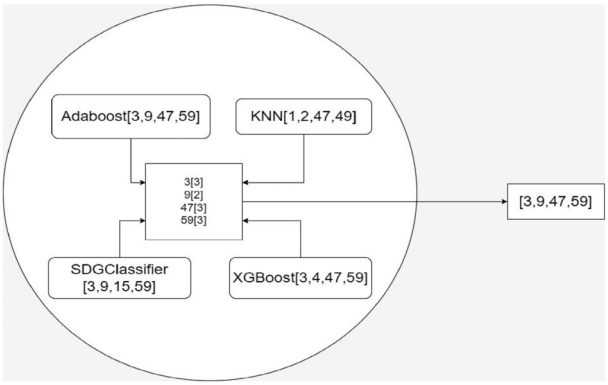


Fig. 2. Features selected for NSLKDD dataset.

Classifier	Features extracted	Features selected
AdaBoost	[3,9,47,59]	[3,9,47,59]
KNN	[1,2,47,49]	[47]
SDGClassifier	[3,9,15,59]	[3,9,59]
XGBoost	[3,4,47,59]	[3,47,59]

Table 1. Feature selection for NSL-KDD.

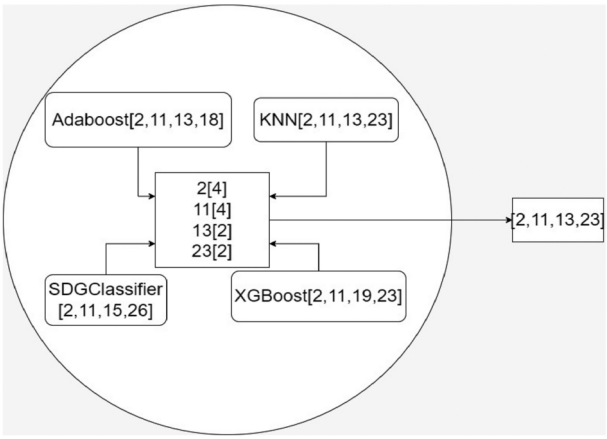


Fig. 3. Features selected for CICIDS dataset.

Classifier	Features extracted	Features selected
AdaBoost	[2,11,13,18]	[2,11,13]
KNN	[2,11,13,23]	[2,11,13,23]
SDGClassifier	[2,11,15,26]	[2,11]
XGBoost	[2,11,19,23]	[2,11,23]

Table 2. Feature selection for CIC-IDS 2017.

SDG [2,3,15,26], and XGBOOST [2,5,19,24] out of these we have selected the feature subset [2,3,15,24] which helps in the classification process.

To get the best features feasible, exhaustive feature selection employs the classification methods perception including Adaboost, SDG, XGBoost, and KNN classifiers. The most precise classifier is selected based on the accuracy of its results. A subset of the pertinent features of this best classifier is considered. To prevent clones, we

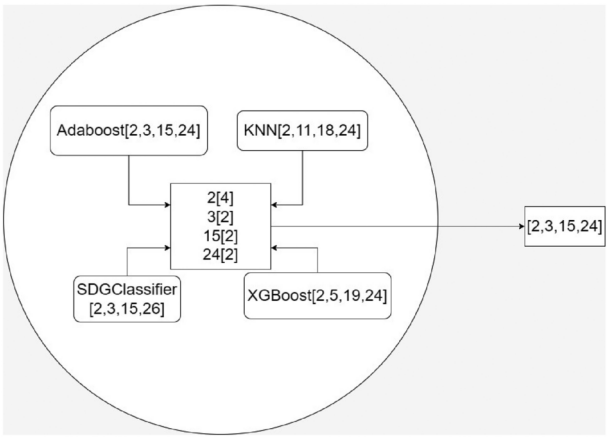


Fig. 4. Features selected for UNSW-NB15 dataset.

Classifier	Features extracted	Features selected
AdaBoost	[2,3,5,24]	[2,3,24]
KNN	[2,11,18,24]	[2,24]
SDGClassifier	[2,3,15,24]	[2,3,15,24]
XGBoost	[2,5,19,24]	[2,24]

Table 3. Feature selection for UNSW-NB15 dataset.

used clone estimators. After fitting the data with the x train and y train, we pass it to the LS-SVM to do further classification.

For the NSL-KDD dataset, the accuracy of the classifiers in the exhaustive feature selection is 97%, 88% , 83%, and 95%, and the best features selected are (3,9,47,59). For the CIC-IDS dataset. The accuracy is in the exhaustive feature selection 94%, 82%, 96%, and 95% and the best features selected are (2,11,13,23). For the UNSW-NB15 dataset, the accuracy of the classifiers in the exhaustive feature selection is 98%, 95%, 84%, and 98%, and the best features selected are (2,3,15,24).

Model classification based on LS-SVM

Least-Squares SVM (LS-SVM) provides a mathematical framework that simplifies the standard SVM optimization by converting it into a set of linear equations. In classical SVM, one minimizes a hinge-loss with inequality constraints, leading to a quadratic programming problem. By contrast, LS-SVM uses a quadratic loss (sum of squared errors) with equality constraints, yielding a linear Karush–Kuhn–Tucker (KKT) system that is much easier to solve. This property is central to both classical and quantum-inspired acceleration of SVM. Indeed, a quantum algorithm proposed by Rebentrost et al. achieved an exponential speedup for solving the LS-SVM optimization by operating in a high-dimensional Hilbert space and using quantum linear system solvers. Inspired by that quantum approach, Ding et al.[14] developed a *quantum-inspired* LS-SVM algorithm that retains the same mathematical formulation as classical LS-SVM but employs randomized techniques (instead of actual quantum computing) to solve it efficiently. We next detail the LS-SVM formulation and highlight how the quantum-inspired method leverages it.

Consider a binary classification task with training set $\{(x_i, y_i)\}_{i=1}^m$, where $x_i \in \mathbb{R}^n$ represents the feature vector and $y_i \in \{+1, -1\}$ denotes the corresponding class label for the *i*-th sample. The Least-Squares Support Vector Machine (LS-SVM) aims at finding a decision function of the form:

$$f(x) = w^T \varphi(x) + b = \sum_{i=1}^{m} \alpha_i K(x_i, x) + b, \tag{1}$$

where $\varphi(x)$ represents a potentially nonlinear mapping from the original input space into a higher-dimensional feature space, designed to facilitate linear separation of data points. Here, w is the weight vector in the mapped feature space, and b is the bias term. The function $K(x_i, x) = \varphi(x_i)^T \varphi(x)$ is known as the kernel function, which implicitly performs this mapping without explicitly computing $\varphi(x)$ itself. Common examples of kernel functions include the polynomial kernel $K(x_i, x_j) = (x_i^T x_j + c)^d$ and the Gaussian (RBF) kernel defined as $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$, where parameters c, d, and σ define the kernel's specific behavior. Equation (1) illustrates that once the parameters $\{\alpha_i\}$ and the bias b are determined, the class of a new input sample x can be predicted using the sign of $f(x)$.

When using a linear kernel or implicitly handling the nonlinear mapping $\varphi(x_i)$, the primal optimization problem of LS-SVM involves minimizing an objective function that balances margin maximization and classification error minimization. Formally, the problem is defined as:

$$\min_{w,e} \quad \frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\sum_{i=1}^{m} e_i^2. \tag{2}$$

The term $\frac{1}{2}\|w\|^2$ regularizes the model by maximizing margin width through minimizing the weight vector w, thus improving generalization. The second term, $\frac{\gamma}{2}\sum_{i=1}^{m} e_i^2$, penalizes classification errors represented by slack variables e_i. Here, $\gamma > 0$ is a regularization parameter controlling the trade-off between margin width and error penalty.

The optimization is subject to equality constraints, ensuring each training sample adheres explicitly to margin conditions:

$$y_i(w^T x_i + b) = 1 - e_i, \quad i = 1, \ldots, m. \tag{3}$$

These constraints enforce precise margin conditions: a slack variable $e_i = 0$ indicates that the sample lies exactly on the margin, $e_i > 0$ means the sample crosses into the margin or is misclassified, and $e_i < 0$ indicates samples within the margin boundary. Thus, e_i quantifies both the presence and severity of classification errors.

To solve the optimization problem defined by Eqs. (2) and (3), the Lagrangian formulation is introduced. This converts the constrained optimization into an unconstrained one by integrating the constraints using Lagrange multipliers α_i. The resulting Lagrangian for the LS-SVM problem is expressed as:

$$\begin{aligned}\mathcal{L}(w, b, e, \alpha) = &\frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\sum_{i=1}^{m} e_i^2 \\ &- \sum_{i=1}^{m} \alpha_i \left[y_i(w^T \varphi(x_i) + b) - 1 + e_i \right],\end{aligned} \tag{4}$$

where $\alpha_i \in \mathbb{R}$ are the Lagrange multipliers. These multipliers enforce the constraints from (3), connecting margin conditions explicitly to the primal variables. Minimizing Eq. (4) with respect to w, b, e_i, and α_i simplifies the derivation of the Karush-Kuhn-Tucker (KKT) optimality conditions, streamlining the calculation of the optimal solution.

By taking partial derivatives of the Lagrangian in Eq. (4) with respect to w, b, and e_i and setting them equal to zero, we obtain the KKT optimality conditions. These conditions explicitly relate the primal and dual variables as follows:

First, the optimal weight vector w is represented as a linear combination of the mapped feature vectors:

$$w = \sum_{i=1}^{m} \alpha_i y_i \varphi(x_i). \tag{5}$$

Secondly, the optimality condition enforces a balance constraint on the Lagrange multipliers α_i and the class labels y_i:

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{6}$$

Next, the relationship between slack variables e_i and multipliers α_i is directly governed by the regularization parameter γ:

$$\gamma e_i = \alpha_i, \quad i = 1, \ldots, m. \tag{7}$$

Finally, each training sample must fulfill the primal margin constraints. This margin constraint explicitly connects the slack variable to the sample's location relative to the margin:

$$y_i(w^T \varphi(x_i) + b) = 1 - e_i, \quad i = 1, \ldots, m. \tag{8}$$

By substituting the KKT conditions from Eqs. (2), (3), and (4), one eliminates the primal variables w and slack variables e_i. This substitution transforms the optimization into a simplified linear system involving only the dual variables α_i and bias term b. Formally, this simplified system can be expressed compactly in matrix form as:

$$\begin{pmatrix} 0 & \mathrm{y}^T \\ \mathrm{y} & K + \frac{1}{\gamma} I \end{pmatrix} \begin{pmatrix} b \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{9}$$

where K is the kernel matrix with entries $K_{ij} = K(x_i, x_j)$, $\mathrm{y} = [y_1, \ldots, y_m]^T$ is the vector of class labels, I denotes the identity matrix, and 1 is an m-dimensional vector of ones. This simplified linear algebraic system effectively removes primal variables w and slack variables e_i, expressing the solution explicitly in terms of dual variables $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T$ and bias b.

Once the system (9) is solved for the dual variables $\{\alpha_i\}$ and bias b, the LS-SVM decision function for any new sample x becomes:

$$f(x) = \sum_{i=1}^{m} \alpha_i K(x_i, x) + b, \tag{10}$$

where the predicted class of x is determined simply by the sign of $f(x)$ as:

$$y(x) = \mathrm{sgn}[f(x)]. \tag{11}$$

The Least-Squares Support Vector Machine (LS-SVM) approach presented through Eqs. (2), (3), (4), and (9) offers notable advantages for intrusion detection systems (IDS). Primarily, the transformation of the optimization problem into a simplified linear system (Eq. (9)) significantly enhances computational efficiency, making LS-SVM highly scalable to large datasets typically encountered in network environments. Such scalability is critical for IDS applications, where rapid detection and response are paramount.

The LS-SVM formulation inherently balances margin maximization (achieved by minimizing $\|w\|^2$ in Eq. (2)) with the minimization of classification errors (quantified by slack variables e_i). This built-in regularization reduces overfitting, ensuring robust generalization to previously unseen network activity. Coupling LS-SVM with an exhaustive feature selection process further enhances its capability, as selecting optimal and informative features directly improves model accuracy by eliminating redundant or irrelevant information that may introduce noise or lead to overfitting.

Moreover, the kernel-based structure of LS-SVM allows flexibility in modeling complex, nonlinear relationships within network traffic. Adjusting kernel parameters, such as σ in Gaussian kernels or d in polynomial kernels, permits the classifier to adapt effectively to various intrusion patterns, making LS-SVM highly adaptable to dynamic network threats.

Overall, by integrating LS-SVM's computationally efficient linear algebraic optimization with exhaustive feature selection, the resulting IDS is both highly accurate and computationally lightweight, making it ideally suited for real-time threat detection in modern, complex network environments.

Metrics

Accuracy

Accuracy percentage is a measure of how many records in the dataset are classified accurately[47].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{12}$$

where *TP* denotes the True Positive when the model correctly predicts the positive outcome. *FP* is the False Positive when the model incorrectly predicts the positive outcome. *TN* denotes the True Negative when the model correctly predicts the negative outcome. *FN* is the False Negative when the model incorrectly predicts the negative outcome.

Precision

The precision percentage is a measure of how many records in the dataset are predicted accurately[48].

$$Precision = \frac{TP}{TP + FP}. \tag{13}$$

Recall

The *recall* is defined as the ratio of Positive samples that are correctly identified to all Positive samples. The recall gauges how well the model can identify Positive samples. The more positive samples that are correctly identified, the larger the recall[49].

$$Recall = \frac{TP}{TP + FN}. \tag{14}$$

F1-score

The F1-score is important for assessing the model to obtain a complete evaluation. It is a function of recall and precision and is also known as the mean of recall and precision.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{15}$$

Experimental results

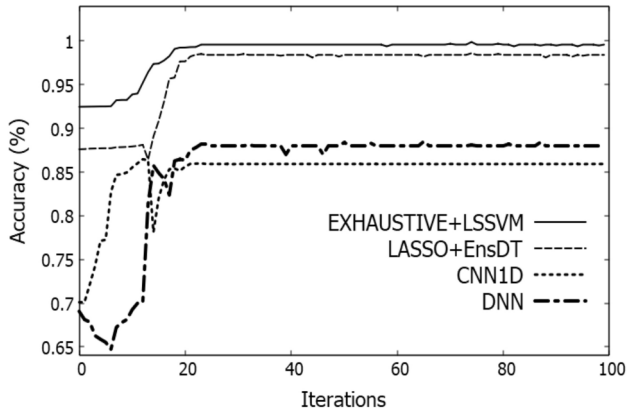
The experiments were carried out using Python and ML sklearn library on computers with the configuration of Intel® CoreTM i5-7200U CPU at 2.70 GHz, 24 GB memory, 64-bit Windows11 operating system, 1 TB SSD and Ryzen 7-8500 CPU at3.20GHz, 16GB memory, Windows 10 operating system, 1TB SSD.

Figures 5, 6, and 7 show the weighted features of the NSL-KDD, CIC-IDS, and UNSW-NB15 datasets where all the features are arranged in descending order regarding the importance of the feature in the dataset before the Exhaustive feature selection was employed. During exhaustive feature selection, all possible features are combined and as a result, the best feature subset is produced.

NSL-KDD dataset

Table 4 shows the training accuracy and the error rate of all the models. Our model has the least error rate of 0.68 which is quite low compared to other models. Accordingly, our model doesn't undergo overfitting compared to other models with high error rates.

Figure 6A shows the training accuracy for various models with the NSL-KDD dataset. Unlike other methods, the accuracy has started at 93% and increases until it reaches 99.3% and sustains the accuracy throughout all the iterations. Whereas, LASSO+EnsDT has an accuracy of 88% initially and gradually increases to 97% later on. On the other hand, CNN1D and DNN struggle at 5 to 10 iterations with accuracy of 85% and 88%, respectively, but cannot maintain it throughout all the iterations.



Keeping track of the training loss is important to improve the model's performance. Figure 8 shows that exhaustive+LS-SVM has the least loss compared to other models in the NSL-KDD dataset. The loss values are 0.7, 0.9, 1.2, and 1.4 for exhaustive+LS-SVM, LASSO+EnsDT, CNN1D, and DNN, respectively. The loss of exhaustive+LS-SVM gradually decreases throughout the iterations whereas the other models experience fluctuations and struggle with the data and this results in overfitting and underfitting issues of data. Table 5 shows the loss values for different models implemented.

Figure 9 shows validation metrics of all models implemented with the NSL-KDD dataset. Exhaustive+LS-SVM outperforms other models in all metrics.

These metrics are presented in Table 6 with the NSL-KDD dataset. The exhaustive feature selection method opts for the best subsects of the features with the highest accuracy in the dataset by going through all the possible combinations of the features available. In quantum-inspired LS-SVM, the input data is mapped into a higher-dimensional space to improve accuracy. Therefore, by implementing LS-SVM on the best subsets of features, our model produces the highest efficiency comparatively.

In Table 6, our model gains a higher accuracy of 2.3%, 14.3%, and 11.3% compared to the LASSO ensemble decision tree, CNNID, and DNN, respectively.

CIC-IDS 2017 dataset

Table 7 shows the training accuracy and the error rate of all the models with the CIC-IDS dataset. It is obvious that our model has the least error rate of 0.42 which is quite low compared to other models. Accordingly, our model doesn't suffer from overfitting compared to other models with high error rates.

Figure 10 shows the training accuracy for various models with the CIC-IDS dataset. Dissimilar to other methods, the accuracy starts at 88% and increases until it reaches 99.58% and sustains the accuracy throughout all the iterations. Nevertheless, LASSO+EnsDT has an accuracy of 70% initially and gradually increases to 88.42% ultimately. Furthermore, CNN1D and DNN struggle at 5 to 10 iterations with accuracy of 82% and 87%, respectively, but cannot maintain it throughout all the iterations.

Reducing the training loss is crucial to enhance the model performance. Figure 11 shows that exhaustive+LS-SVM has the least loss compared to other models with the CIC-IDS dataset. The loss values are 0.7, 0.9, 1.1, and 1.4 for exhaustive+LS-SVM, LASSO+EnsDT, CNN1D, and DNN, respectively. The loss of exhaustive+LS-SVM gradually decreases throughout the iterations whereas the other models experience fluctuations and struggle with the data and this results in overfitting and underfitting issues of data. Table 8 shows the loss values for different models implemented with the CIC-IDS dataset.

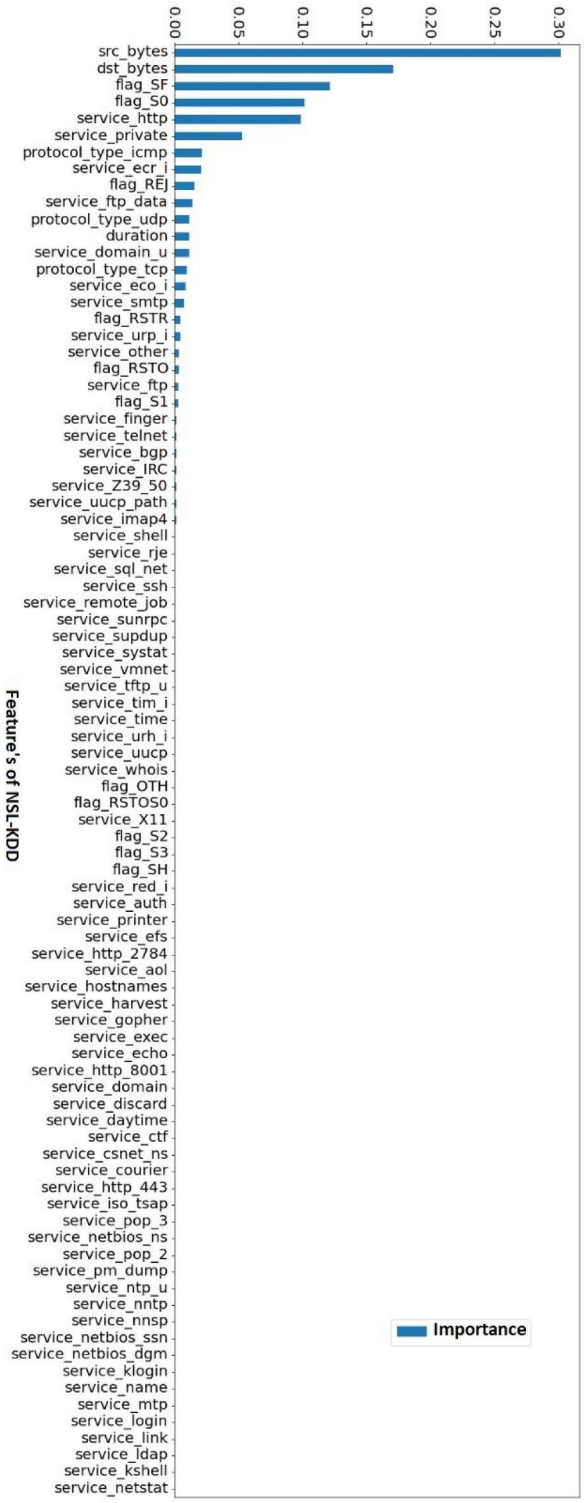


Fig. 5. Weighted features of NSL-KDD dataset.

Figure 12 shows validation metrics of all models implemented with the CIC-IDS 2017 dataset. Exhaustive+LS-SVM performs better than other models in all metrics as presented in Table 9. The exhaustive feature selection method picks out the best subsects of the features with the highest accuracy in the dataset by going through all the possible combinations of the features available. In quantum-inspired LS-SVM, the input data is mapped into a higher-dimensional space to enhance the accuracy. Accordingly, by implementing LS-SVM on the best subsets of features, our model results in the highest efficiency comparatively.

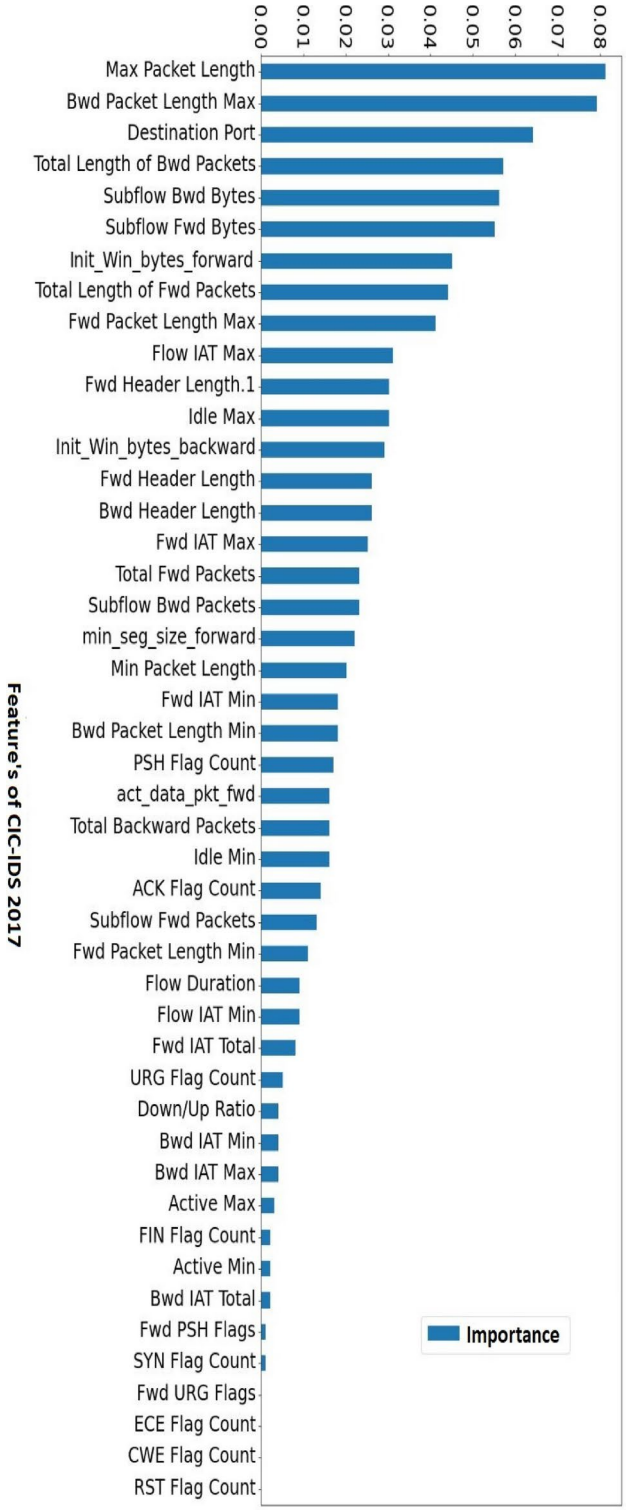


Fig. 6. Weighted features of CIC-IDS dataset.

In Table 9, our model gains a higher accuracy of 11%, 17.5%, and 12.5% compared to the LASSO ensemble decision tree, CNNID, and DNN, respectively.

UNSW-NB15 dataset

Table 10 shows the training accuracy and the error rate of all the models with the UNSW-NB15 dataset. Our model has the least error rate of 6.73 which is quite low compared to other models. Accordingly, our model doesn't experience overfitting compared to other models with high error rates.

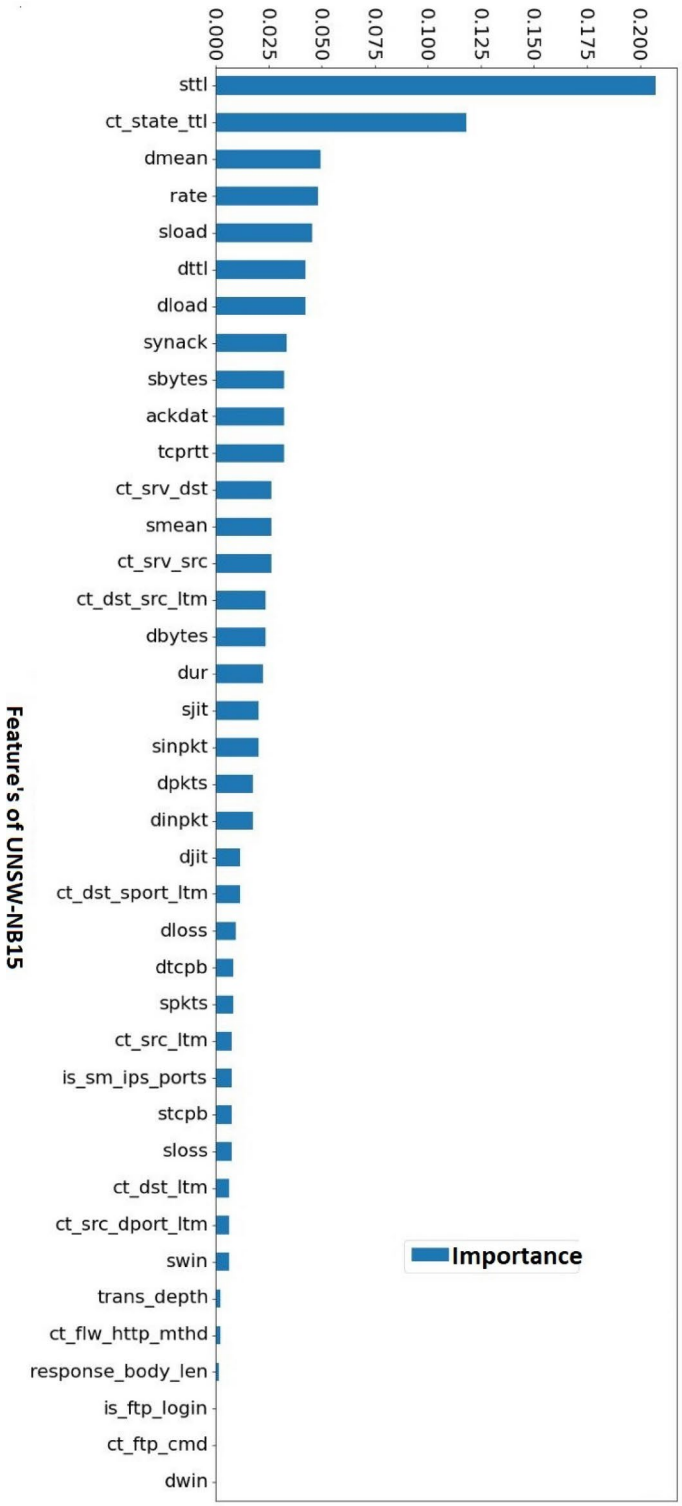


Fig. 7. Weighted features of UNSW-NB15 dataset.

Figure 13 shows the training accuracy for various models with the UNSW-NB15 dataset. The accuracy of our model starts at 73.3% and increases until it reaches 93.27% and sustains the accuracy throughout all the iterations. Nevertheless, LASSO+EnsDT has an accuracy of 70% initially and gradually increases to 76.7% ultimately. In addition, the accuracy of CNN1D reaches 86% while DNN fluctuates around 87% with iterations.

As indicated earlier, lowering the training loss is pivotal to ameliorating the model performance. Figure 14 shows that with the UNSW-NB15 dataset, our model has the minimum loss with limited fluctuations with

Model	Accuracy (%)	Error rate
EFS + LS-SVM	99.32%	0.68
RFE + EnsDT[7]	97%	3
CNNID[50]	85%	15
DNN[51]	88%	12

Table 4. Training accuracy and error rate for various models with NSL-KDD dataset.

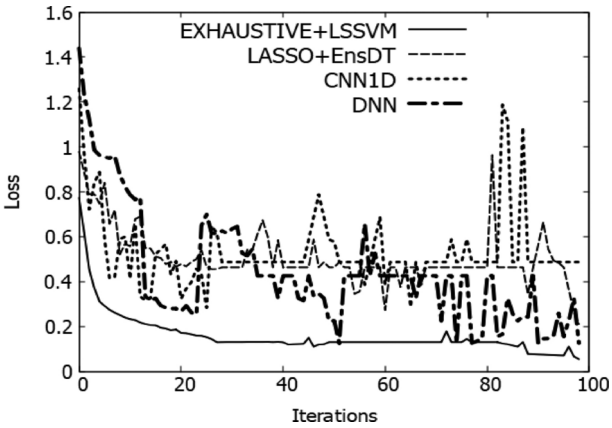


Fig. 8. Training loss with NSL-KDD dataset.

Model	Loss
EFS + LS-SVM	0.7
LASSO + EnsDT	0.9
CNNID	1.2
DNN	1.4

Table 5. Training loss with NSL-KDD dataset.

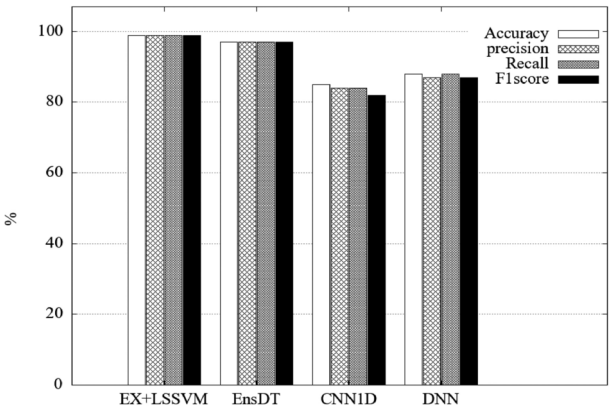


Fig. 9. Validation metrics for all models with NSL-KDD dataset.

iterations while other models have a higher loss and substantial fluctuations. Table 11 shows the loss values for different models implemented with the UNSW-NB15 dataset.

As pointed out earlier with other datasets, our model with the UNSW-NB15 dataset does not undergo overfitting as the loss is minimal compared to other models.

Figure 15 shows validation metrics of all models implemented with the UNSW-NB15 dataset. Exhaustive+LS-SVM performs better than other models in all metrics. These metrics are presented in Table 12. Our model

Model	Accuracy	Precision	Recall	F1-score
EFS + LS-SVM	99.3%	0.99	0.99	0.99
LASSO + EnsDT	97%	0.97	0.97	0.97
CNNID	85%	0.84	0.84	0.82
DNN	88%	0.87	0.88	0.87

Table 6. Evaluation metrics for all models with NSL-KDD dataset.

Model	Accuracy (%)	Error rate
EFS + LS-SVM	99.58%	0.42
RFE + EnsDT[7]	88.42%	11.58
CNNID[50]	82%	18
DNN[51]	87%	13

Table 7. Training accuracy and error rate for various models with CIC-IDS 2017 dataset.

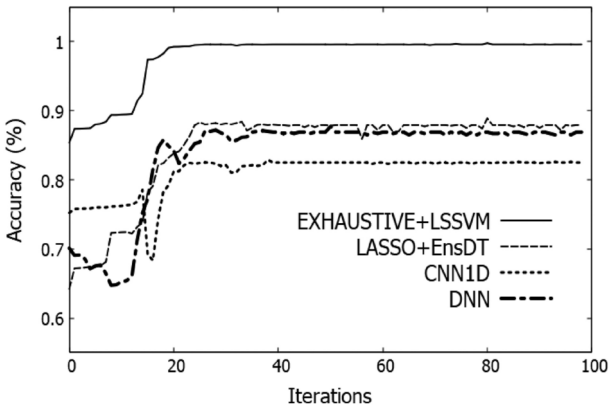


Fig. 10. Training accuracy with CIC-IDS 2017 dataset.

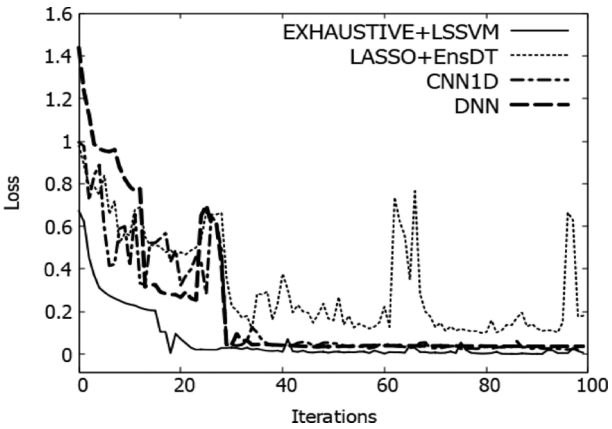


Fig. 11. Training loss with CIC-IDS 2017 dataset.

behaves with the UNSW-NB15 dataset similarly as with other datasets where the exhaustive feature selection method picks out the best subsects of the features with the highest accuracy in the dataset by going through all the possible combinations of the features available.

In Table 12, our model gains a higher accuracy of 16.57%, 7.27%, and 6.27% compared to the LASSO ensemble decision tree, CNNID, and DNN, respectively.

Model	Loss
EFS + LS-SVM	0.7
LASSO + EnsDT	0.9
CNNID	1.1
DNN	1.4

Table 8. Training loss with CIC-IDS 2017 dataset.

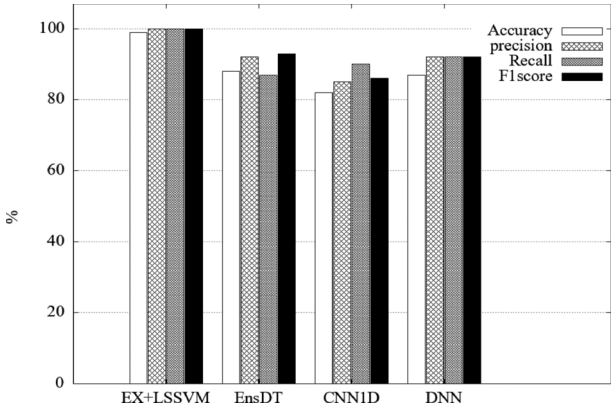


Fig. 12. Validation metrics for all models with CIC-IDS 2017 dataset.

Model	Accuracy	Precision	Recall	F1-score
EFS + LS-SVM	99.5%	1.00	1.00	1.00
EFS + LS-SVM	99.5%	1.00	1.00	1.00
LASSO + EnsDT	88.42%	1.00	0.87	0.91
CNNID	82%	0.85	0.90	0.86
DNN	87%	0.92	0.92	0.92

Table 9. Evaluation metrics for all models with CIC-IDS 2017 dataset.

Model	Accuracy (%)	Error rate
EFS + LS-SVM	93.27%	6.73
RFE + EnsDT[7]	76.7%	13.3
CNNID[50]	86%	14
DNN[51]	87%	13

Table 10. Training accuracy and error rate for various models with UNSW-NB15 dataset.

Model analysis

Our model employs a comprehensive feature selection approach designed to determine the most effective subsets of features for classification, regardless of dataset size or imbalance. By evaluating every possible combination of features, we ensure that the classifier achieves optimal accuracy. It is important that the Intrusion Detection System (IDS) not depend solely on a single dataset during training or testing, as this could lead to misclassifications for new or evolving types of attacks. For example, while the LASSO+EnsDT model may achieve high accuracy on the NSL-KDD dataset, it may deteriorate in performance when tested on other datasets, such as CIC-IDS-2017 or UNSW-NB15. In contrast, our proposed system demonstrates consistently high accuracy across all three datasets, thus reflecting its robustness against diverse intrusion scenarios. Although accuracy is a key indicator of performance, it is insufficient on its own-particularly for imbalanced datasets. Consequently, we emphasize additional metrics such as precision and recall. Our system demonstrates negligible false positives (FP) and false negatives (FN) compared to true positives (TP) and true negatives (TN), meaning that both benign and malicious traffic are accurately identified for all three datasets. Further, in scenarios with skewed class distributions, F1-Score often provides a better measure of performance by balancing precision and recall. We

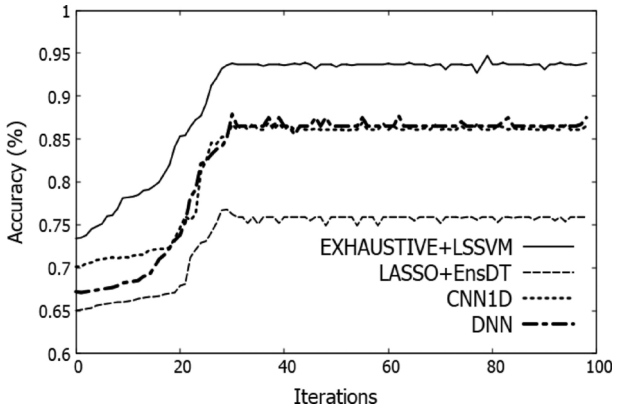


Fig. 13. Training accuracy with UNSW-NB15 dataset.

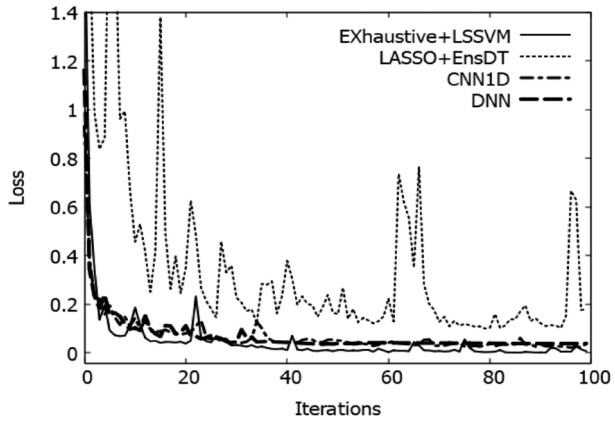


Fig. 14. Training loss with UNSW-NB15 dataset.

Model	Loss
EFS + LS-SVM	0.6
LASSO + EnsDT	1.4
CNNID	1.00
DNN	1.4

Table 11. Training loss with UNSW-NB15.

observe that our system achieves excellent F1-Scores in such cases, illustrating its effectiveness at distinguishing threats from benign traffic. In the following, we provide thorough studies to show the effectiveness of our model.

Ablation study

An ablation study was conducted to measure the individual and combined impacts of two main components in our proposed IDS: the exhaustive feature selection algorithm and the quantum-inspired Least Square Support Vector Machine (LS-SVM) classifier. By selectively adding or removing these components, we measured changes in accuracy, precision, recall, and F1-score, thereby isolating each component's contribution to overall performance.

Effect of feature selection across datasets

We first assessed the role of Exhaustive Feature Selection (EFS) on three benchmark datasets: NSL-KDD, CIC-IDS-2017, and UNSW-NB15. Table 13 summarizes the results for our LS-SVM-based classifier. In all cases, EFS boosted accuracy, precision, recall, and F1-score significantly, underscoring that reducing irrelevant or redundant features enhances predictive performance. Notably, accuracy for NSL-KDD jumps from 85.0 to 99.3%, while recall and precision both climb above 0.98%. Similar improvements occur in the CIC-IDS-2017 and UNSW-NB15 datasets.

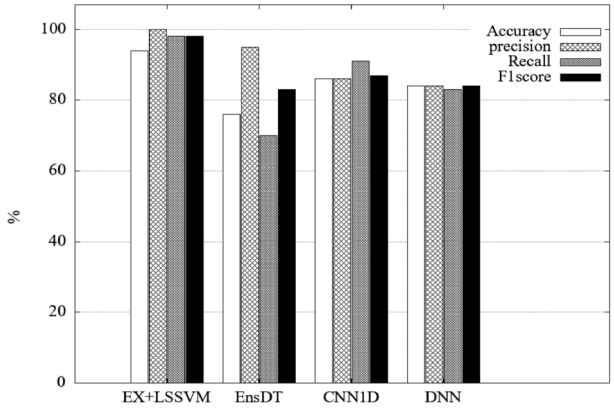


Fig. 15. Validation metrics for all models with UNSW-NB15 dataset.

Model	Accuracy	Precision	Recall	F1-score
EFS + LS-SVM	93.27%	1.00	0.98	0.98
LASSO + EnsDT	76.70%	1.00	0.70	0.83
CNNID	86%	0.86	0.91	0.87
DNN	87%	0.84	0.83	0.84

Table 12. Evaluation metrics for all models with UNSW-NB15 dataset.

Dataset	Model Variant	Accuracy (%)	Precision	Recall	F1-score
NSL-KDD	Without EFS	85.0	0.84	0.84	0.82
	With EFS	99.3	0.99	0.99	0.99
CIC-IDS-2017	Without EFS	82.0	0.85	0.84	0.83
	With EFS	99.5	1.00	1.00	1.00
UNSW-NB15	Without EFS	78.0	0.79	0.80	0.78
	With EFS	93.3	1.00	0.98	0.98

Table 13. Performance with and without exhaustive feature selection (EFS) across all datasets for LS-SVM.

Comparison of classifiers across datasets

We next compared multiple models with classifiers-AdaBoost, KNN, SGD, XGBoost, LASSO+EnsDT, CNNID, DNN, and our proposed EFS+LS-SVM-across the same three datasets. Table 14 presents these results without feature selection, while Table 15 captures the same classifiers when EFS is used. In both tables, EFS+LS-SVM consistently achieves the highest accuracy, precision, recall, and F1-score. Although AdaBoost and XGBoost also demonstrate relatively strong performance, the inclusion of exhaustive feature selection combined with LS-SVM yields superior results.

The ablation study thus highlights the essential nature of both exhaustive feature selection and the LS-SVM classifier in delivering robust, high-accuracy intrusion detection. Feature selection filters out superfluous features, reducing overfitting and improving classification efficiency, while LS-SVM provides a reliable and scalable means of distinguishing between malicious and benign network traffic.

Runtime analysis

In addition to classification performance, we evaluated the efficiency of each classifier by measuring training and testing runtimes on NSL-KDD, CIC-IDS-2017, and UNSW-NB15. Table 16 provides runtime data *without* feature selection, while Table 17 shows runtime metrics for models *with* exhaustive feature selection.

A direct comparison between these tables reveals that EFS substantially reduces both training and testing times for nearly every classifier. For instance, LS-SVM's training time on NSL-KDD rises from 3.0 s (with EFS) to 4.2 s (without EFS), and the total time increases proportionally. Similar patterns appear for AdaBoost, KNN, SGD, and XGBoost. The largest overheads tend to occur in high-dimensional datasets such as CIC-IDS-2017, emphasizing the value of feature selection in mitigating computational costs.

Computational efficiency of classifiers

Among all evaluated classifiers, LS-SVM offers a strong balance between high accuracy and efficient runtime, emerging as a promising candidate for real-time IDS deployments. Meanwhile, AdaBoost achieves high

Dataset	Classifier	Accuracy (%)	Precision	Recall	F1-score
NSL-KDD	AdaBoost	90.0	0.90	0.90	0.90
	KNN	80.0	0.80	0.81	0.80
	SGD	78.0	0.78	0.78	0.78
	XGBoost	88.0	0.88	0.88	0.88
	LS-SVM	85.0	0.85	0.85	0.85
	LASSO + EnsDT	90.0	0.90	0.90	0.90
	CNNID	78.0	0.78	0.78	0.78
	DNN	80.0	0.80	0.80	0.80
CIC-IDS-2017	AdaBoost	80.0	0.92	0.85	0.88
	KNN	75.0	0.80	0.78	0.80
	SGD	73.0	0.78	0.78	0.78
	XGBoost	83.0	0.88	0.85	0.88
	LS-SVM	82.0	0.82	0.82	0.82
	LASSO + EnsDT	80.0	0.83	0.82	0.83
	CNNID	74.0	0.78	0.78	0.78
	DNN	77.0	0.80	0.80	0.80
UNSW-NB15	AdaBoost	70.0	0.95	0.68	0.78
	KNN	80.0	0.80	0.85	0.84
	SGD	78.0	0.78	0.80	0.80
	XGBoost	82.0	0.85	0.83	0.84
	LS-SVM	78.0	0.78	0.78	0.78
	LASSO + EnsDT	70.0	0.95	0.68	0.78
	CNNID	80.0	0.80	0.85	0.84
	DNN	80.0	0.80	0.80	0.80

Table 14. Performance of models without exhaustive feature selection across all datasets.

Dataset	Classifier	Accuracy (%)	Precision	Recall	F1-score
NSL-KDD	AdaBoost	97.0	0.97	0.97	0.97
	KNN	88.0	0.87	0.88	0.87
	SGD	83.0	0.82	0.83	0.82
	XGBoost	95.0	0.95	0.95	0.95
	EFS+LS-SVM	99.3	0.99	0.99	0.99
	LASSO + EnsDT	97.0	0.97	0.97	0.97
	CNNID	85.0	0.84	0.84	0.82
	DNN	88.0	0.87	0.88	0.87
CIC-IDS-2017	AdaBoost	88.4	1.00	0.87	0.91
	KNN	82.0	0.85	0.84	0.83
	SGD	80.0	0.83	0.83	0.81
	XGBoost	90.0	0.92	0.90	0.91
	EFS+LS-SVM	99.5	1.00	1.00	1.00
	LASSO + EnsDT	88.5	0.89	0.88	0.89
	CNNID	81.0	0.84	0.83	0.83
	DNN	84.0	0.86	0.85	0.85
UNSW-NB15	AdaBoost	76.7	1.00	0.70	0.83
	KNN	86.0	0.86	0.91	0.87
	SGD	84.0	0.83	0.84	0.82
	XGBoost	88.0	0.89	0.89	0.89
	EFS+LS-SVM	93.3	1.00	0.98	0.98
	LASSO + EnsDT	76.7	1.00	0.70	0.83
	CNNID	86.0	0.86	0.91	0.87
	DNN	87.0	0.84	0.83	0.84

Table 15. Performance of models with exhaustive feature selection across all datasets.

Classifier	NSL-KDD			CIC-IDS-2017			UNSW-NB15		
	Train (s)	Test (s)	Total (s)	Train (s)	Test (s)	Total (s)	Train (s)	Test (s)	Total (s)
LS-SVM	4.2	3.6	7.8	10.5	1.8	12.3	5.5	3.5	9.0
AdaBoost	9.5	7.2	16.7	28.0	7.8	35.8	14.5	11.2	25.7
KNN	4.8	4.5	9.3	3.5	16.0	19.5	5.5	7.5	13.0
SGD	6.8	4.0	10.8	21.5	3.2	24.7	6.2	3.8	10.0
XGBoost	8.0	5.5	13.5	19.5	11.5	31.0	9.2	3.5	12.7
LASSO + EnsDT	5.3	4.5	9.8	10.5	3.9	14.4	5.9	4.2	10.1
CNNID	16.8	12.6	29.4	42.0	11.2	53.2	25.2	9.8	35.0
DNN	18.2	14.0	32.2	44.8	12.6	57.4	28.0	11.2	39.2

Table 16. Combined runtime metrics for models without exhaustive feature selection.

Classifier	NSL-KDD			CIC-IDS-2017			UNSW-NB15		
	Train (s)	Test (s)	Total (s)	Train (s)	Test (s)	Total (s)	Train (s)	Test (s)	Total (s)
LS-SVM	3.0	2.8	5.8	8.0	1.0	9.0	4.0	2.8	6.8
AdaBoost	7.0	5.6	12.6	25.0	6.5	31.5	12.0	9.0	21.0
KNN	3.5	3.5	7.0	2.0	13.0	15.0	4.0	6.0	10.0
SGD	5.0	3.0	8.0	18.0	2.0	20.0	4.5	2.5	7.0
XGBoost	6.0	4.0	10.0	16.0	9.0	25.0	7.0	2.0	9.0
LASSO + EnsDT	3.8	3.2	7.0	7.5	2.8	10.3	4.2	3.0	7.2
CNNID	12.0	9.0	21.0	30.0	8.0	38.0	18.0	7.0	25.0
DNN	13.0	10.0	23.0	32.0	9.0	41.0	20.0	8.0	28.0

Table 17. Combined runtime metrics for models with exhaustive feature selection.

predictive power but at the cost of longer runtimes, whereas KNN's testing phase scales poorly with increasing dataset size. CNNID and DNN, though competitive in certain scenarios, generally exhibit higher computational overhead due to their complex deep-learning architectures.

- *Exhaustive+LS-SVM* exhibits the lowest overall runtime across all datasets, with an average total execution time of 7.2 s, ensuring rapid classification without significant computational burden.
- *AdaBoost* incurs the highest computational cost, particularly in the CIC-IDS-2017 dataset, where it requires a total of 31.5 seconds. This suggests that while boosting techniques enhance predictive performance, they introduce substantial computational overhead.
- *KNN* shows considerable variation in runtime, with significantly higher test phase durations in datasets with high-dimensional feature spaces. This aligns with the known inefficiencies of KNN, where classification complexity grows with dataset size.
- *SGD* achieves a moderate runtime profile, but its test phase duration remains relatively low, indicating its suitability for applications demanding real-time inference.
- *XGBoost* provides an intermediate balance, demonstrating a stable runtime performance across all datasets.

Practical implications

The runtime metrics underscore the importance of selecting an appropriate classifier based on deployment constraints. In operational IDS settings, where rapid threat detection is imperative, LS-SVM emerges as a preferred choice due to its optimized execution time and high classification accuracy. In contrast, computationally intensive models like AdaBoost and KNN may require optimization strategies, such as parallelization or feature reduction, to improve their feasibility for large-scale deployments.

Furthermore, for practical implications, we should consider real-world IDS environments where rapid and accurate detection of malicious traffic is paramount. Our analysis suggests that EFS+LS-SVM is both accurate and computationally feasible. Models such as AdaBoost or KNN may require hardware acceleration, parallel processing, or additional dimensionality reduction to maintain real-time responsiveness in large-scale or latency-sensitive contexts.

These findings reinforce the efficacy of our proposed system, highlighting its capability to maintain high detection accuracy while ensuring computational efficiency. The runtime analysis further substantiates the robustness of our framework, as it evaluates the performance of models incorporating these classifiers, making it well-suited for real-world intrusion detection applications. Additionally, as demonstrated in Table 17, our model achieves an accuracy of 99.3%, 99.5%, and 93.27% across NSL-KDD, CICIDS2017, and UNSW-NB15, respectively, further validating its effectiveness.

Conclusion

In this work, we present a machine-learning-based intrusion detection system (IDS) that leverages Exhaustive Feature Selection (EFS) to thoroughly evaluate all possible feature subsets and isolate those that maximize classification accuracy. By narrowing down only the most pertinent features, our framework effectively eliminates noise and redundancy, enhancing overall system robustness. The quantum-inspired Least Square Support Vector Machine (LS-SVM) classifier is then employed on these refined features to further elevate detection performance, offering both high accuracy and low false prediction rates while minimizing training overhead.

Empirical evaluation on three benchmark datasets-NSL-KDD, CIC-IDS-2017, and UNSW-NB15-demonstrates the strong effectiveness of the proposed system. Notably, our model attains accuracy scores of 99.3% for NSL-KDD, 99.5% for CIC-IDS-2017, and 93.3% for UNSW-NB15, surpassing many established methods. These high accuracy values are further supported by consistently strong precision and recall metrics. For example, precision holds at 1.00% in CIC-IDS-2017 and UNSW-NB15, while recall reaches up to 1.00% on CIC-IDS-2017, 0.99% on NSL-KDD, and 0.98 on UNSW-NB15, thus producing high F1-scores across all datasets. Such robust metrics indicate that the system not only identifies malicious traffic with minimal false positives but also effectively captures the full spectrum of potential attacks.

Beyond its high accuracy, the LS-SVM classifier exhibits efficiency in runtime, making it suitable for real-time deployment scenarios. Specifically, we observe testing times of 2.8 s for NSL-KDD, 1.0s for CIC-IDS-2017, and 2.8 s for UNSW-NB15, indicating that the system balances speed and performance in a manner conducive to large-scale or latency-sensitive networks. Furthermore, our model requires the minimum training time for all datasets compared to other models. In addition, minimal loss values observed during training confirm that the classifier avoids overfitting, strengthening its reliability in real-world environments.

To further extend the capabilities of this IDS, future research will investigate its generalizability by testing the trained model on different datasets that share overlapping features. We also plan to adapt the framework to both centralized and federated learning contexts, enabling the incorporation of geographically distributed data sources while preserving user privacy. Through these expansions, we aim to bolster the system's adaptability and resilience in a variety of evolving network landscapes, ensuring that the IDS remains effective against both current and emerging cyber threats.

Data availability

All datasets used are available in Refs.[11–13]. The datasets used and/or analyzed during the current study are available from the first two authors at reasonable request at mkanumur@lakeheadu.ca and pwaghmod@lakeheadu.ca.

Received: 17 September 2024; Accepted: 24 March 2025
Published online: 08 April 2025

References

1. *Network Attacks and Network Security Threats. Cynet*. https://www.cynet.com/network-attacks/network-attacks-and-network-security-threats/ (Accessed 22 November 2022) (2022).
2. Das, S. et al. Network intrusion detection and comparative analysis using ensemble machine learning and feature selection. *IEEE Trans. Netw. Serv. Manag.* **1**, 1. https://doi.org/10.1109/TNSM.2021.3138457 (2021).
3. Terra, J. *10 Years of Artificial Intelligence and Machine Learning, Simplilearn.com. Simplilearn*. https://www.simplilearn.com/ten-years-of-artificial-intelligence-and-machine-learning-article (Accessed 23 December 2022) (2022).
4. Ghelani, J., Gharia, P. & El-Ocla, H. Gradient monitored reinforcement learning for jamming attack detection in FANETs. *IEEE Access* **12**, 23081–23095. https://doi.org/10.1109/ACCESS.2024.3361945 (2024).
5. Brownlee, J. *How to Choose a Feature Selection Method for Machine Learning, MachineLearningMastery.com*. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/ (Accessed 23 December 2022) (2020).
6. *What is Feature Selection? Definition and FAQs | HEAVY.AI*. https://www.heavy.ai/technical-glossary/feature-selection (Accessed 22 November 2022).
7. Croak, M. *A Decade in Deep Learning, and What's Next, Google. Google*. https://blog.google/technology/ai/decade-deep-learning-and-whats-next/ (Accessed 23 December 2022) (2021).
8. Javaid, A. et al. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS)* (2016).
9. Rapid. *The Pros & Cons of Intrusion Detection Systems: Rapid7 blog, Rapid7. Rapid7 Blog*. https://www.rapid7.com/blog/post/2017/01/11/the-pros-cons-of-intrusion-detection-systems/ (2020).
10. Ingre, B. & Yadav, A. Performance analysis of NSL-KDD dataset using ANN. *Int. Conf. Signal Process. Commun. Eng. Syst.* **2015**, 92–96. https://doi.org/10.1109/SPACES.2015.7058223 (2015).
11. *Search UNB (No Date) University of New Brunswick est.1785*. https://www.unb.ca/cic/datasets/nsl.html (Accessed 9 December 2022).
12. Stiawan, D. et al. CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access* **8**, 132911–132921 (2020).
13. Kumar, V. et al. Statistical analysis of the UNSW-NB15 dataset for intrusion detection. In *Computational Intelligence in Pattern Recognition* 279–294 (Springer, 2020).
14. Ding, C., Bao, T.-Y. & Huang, H.-L. Quantum-inspired support vector machine. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(12), 7210–7222. https://doi.org/10.1109/TNNLS.2021.3084467 (2022).
15. Priyalakshmi, V. & Devi, R. A hybrid framework for effective intrusion detection system in wireless networks. In *International Conference on Computing, Power, and Communication Technologies (IC2PCT)* (2024).
16. Saheed, Y. K., Kehinde, T. O., Ayobami Raji, M. & Baba, U. A. Feature selection in intrusion detection systems: A new hybrid fusion of bat algorithm and residue number system. *J. Inf. Telecommun.* **8**(2), 189–207. https://doi.org/10.1080/24751839.2023.2272484 (2023).
17. Abdulganiyu, O. H. et al. XIDINTFL-VAE: XGBoost-based intrusion detection of imbalance network traffic via class-wise focal loss variational autoencoder. *J. Supercomput.* **81**, 16. https://doi.org/10.1007/s11227-024-06552-5 (2025).
18. Siddiqi, M. A. & Pak, W. Tier-based optimization for synthesized network intrusion detection system. *IEEE Access* **10**, 108530–108544. https://doi.org/10.1109/ACCESS.2022.3213937 (2022).

19. Shah, A. *Through the Eyes of Gabor Filter, Medium. Medium*. https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97 (Accessed 4 January 2023) (2018).
20. Das, S. et al. Network intrusion detection and comparative analysis using ensemble machine learning and feature selection. *IEEE Trans. Netw. Serv. Manag.* **1**, 457. https://doi.org/10.1109/TNSM.2021.3138457 (2021).
21. Chen, Y. et al. Privacy-preserving multi-class support vector machine model on medical diagnosis. *IEEE J. Biomed. Health Inform.* **26**(7), 3342–3353. https://doi.org/10.1109/JBHI.2022.3157592 (2022).
22. Chen, P., Li, F. & Li, J. Research on intrusion detection model based on bagged tree. In *2021 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)* 579–582. https://doi.org/10.1109/TOCS53301.2021.9688618 (2021).
23. Liu, L., Wang, P., Lin, J. & Liu, L. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. *IEEE Access* **9**, 7550–7563. https://doi.org/10.1109/ACCESS.2020.3048198 (2021).
24. Bedi, P., Gupta, N. & Jindal, V. I-SiamIDS: An improved siam-IDS for handling class imbalance in network-based intrusion detection systems. *Int. J. Speech Technol.* **51**(2), 1133–1151 (2021).
25. Brownlee, J. *Extreme Gradient Boosting (XGBoost) Ensemble in Python, MachineLearningMastery.com*. https://machinelearningmastery.com/extreme-gradient-boosting-ensemble-in-python/ (Accessed 23 December 2022) (2021).
26. Verma, A. & Ranga, V. On evaluation of network intrusion detection systems: Statistical analysis of CIDDS-001 dataset using machine learning techniques. *Pertanika J. Sci. Technol.* **26**(3), 1307–1332 (2018).
27. Subba, B., Biswas, S. & Karmakar, S. A neural network based system for intrusion detection and attack classification. *Twenty Second Natl. Conf. Commun.* **2016**, 1–6. https://doi.org/10.1109/NCC.2016.7561088 (2016).
28. Kocher, G. & Kumar, G. A hybrid deep learning approach for effective intrusion detection systems using spatial–temporal features. *Adv. Eng. Sci.* **54**(2), 1503–1519 (2022).
29. Kocher, G. & Kumar, G. Neural network-based hybrid feature extraction method for network intrusion detection systems. *NeuroQuantology* **20**(8), 427–444 (2022).
30. Kumar, S., Gupta, S. & Arora, S. Research trends in network-based intrusion detection systems: A review. *IEEE Access* **9**, 157761–157779. https://doi.org/10.1109/ACCESS.2021.3129775 (2021).
31. Mazini, M., Shirazi, B. & Mahdavi, I. Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *J. King Saud Univ. Comput. Inf. Sci.* **31**, 541–553 (2019).
32. Jyothsna, V. & Prasad, K. M. Anomaly-based intrusion detection system. In *Computer and Network Security*. https://www.intechopen.com/chapters/67618, https://doi.org/10.5772/intechopen.82287 (IntechOpen, 2019).
33. Ozkan-Okay, M., Samet, R., Aslan, Ö. & Gupta, D. A comprehensive systematic literature review on intrusion detection systems. *IEEE Access* **9**, 157727–157760. https://doi.org/10.1109/ACCESS.2021.3129336 (2021).
34. Otoum, Y. & Nayak, A. AS-IDS: Anomaly and signature based IDS for the internet of things. *J. Netw. Syst. Manag.* **29**, 6. https://doi.org/10.1007/s10922-021-09589-6 (2021).
35. Manokaran, J., Vairavel, G. & Vijaya, J. A novel set theory rule based hybrid feature selection techniques for efficient anomaly detection system in IoT edge. In *2023 International Conference on Quantum Technologies, Communications, Computing, Hardware and Embedded Systems Security (iQ-CCHESS), KOTTAYAM, India* 1–6. https://doi.org/10.1109/iQ-CCHESS56596.2023.10391717 (2023).
36. Nabi, F. & Zhou, X. Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security. *Cyber Secur. Appl.* **2**, 1 (2024).
37. Psychogyios, K. et al. Deep learning for intrusion detection systems (IDSs) in time series data. *Future Internet* **16**, 73 (2024).
38. Santos, R. J., Bernardino, J. & Vieira, M. Approaches and challenges in database intrusion detection. *ACM SIGMOD Rec.* **43**(3), 36–47 (2014).
39. Aleesa, A. M., Zaidan, B. B., Zaidan, A. A. & Sahar, N. M. Review of intrusion detection systems based on deep learning techniques: Coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. *Neural Comput. Appl.* **32**(14), 9827–9858 (2020).
40. Ozkan-Okay, M. & Samet, R. Hybrid intrusion detection approach for wireless local area network. In *Proc. 7th Int. Conf. Control Optim. Ind. Appl. (COIA)* 311–313 (2020).
41. Hadem, P., Saikia, D. K. & Moulik, S. An SDN-based intrusion detection system using SVM with selective logging for IP traceback. *Comput. Netw.* **191**, 108015 (2021).
42. Tavallaee, M., Bagheri, E., Lu, W. & Ghorbani, A. A detailed analysis of the KDD CUP 99 data set. In *Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)* (2009).
43. *Search UNB (No Date) University of New Brunswick est.1785*. https://www.unb.ca/cic/datasets/ids-2017.html (Accessed 9 December 2022).
44. *The UNSW-NB15 Dataset | UNSW Research*. https://research.unsw.edu.au/projects/unsw-nb15-dataset (Accessed 9 December 2022).
45. Moustafa, N. & Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *Milit. Commun. Inf. Syst. Conf.* **2015**, 1–6. https://doi.org/10.1109/MilCIS.2015.7348942 (2015).
46. Potla, T. Scalable machine learning algorithms for big data analytics: Challenges and opportunities. *J. Art. Int. Res.* **2**(2), 124–141 (2022).
47. Alarfaj, F. K. et al. Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access* **10**, 39700–39715. https://doi.org/10.1109/ACCESS.2022.3166891 (2022).
48. Gibson, S., Issac, B., Zhang, L. & Jacob, S. M. Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. *IEEE Access* **8**, 187914–187932. https://doi.org/10.1109/ACCESS.2020.3030751 (2020).
49. Nabipour, M. et al. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access* **8**, 150199–150212. https://doi.org/10.1109/ACCESS.2020.3015966 (2020).
50. Yue, Y., Chen, X., Han, Z., Zeng, X. & Zhu, Y. Contrastive learning enhanced intrusion detection. *IEEE Trans. Netw. Serv. Manag.* **1**, 843. https://doi.org/10.1109/TNSM.2022.3218843 (2022).
51. Chen, X., Han, Z., Zeng, X. & Zhu, Y. Contrastive learning enhanced intrusion detection. *IEEE Trans. Netw. Serv. Manag.* **1**, 843. https://doi.org/10.1109/TNSM.2022.3218843 (2022).

Author contributions

M. K. and P. W. worked on software and writing first draft. H. E. worked on supervision and editing final version of the manuscript. All authors works on paper methodology. T. B. worked with H. E. on responding to the reviewer in the second round of revision.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.E.-O.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025