Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection

[1]Kranthi Kumari Turpati, [2]Mrs.K.U.V.Padma
[1]M.Tech, Dadi Institute of Engineering and Technology
Kranthiturpati1@gmail.com
[2]Assistant Professor, Dadi Institute of Engineering and Technology
kuvpadma@diet.edu.in

Abstract
In security tools like adaptive security appliances, intrusion detection systems, intrusion prevention systems, and firewalls, intrusion detection plays a crucial role. Various techniques are employed for intrusion detection, but their performance is a concern. The effectiveness of intrusion detection relies on accuracy, which must be improved to reduce false alarms and increase the detection rate. To address performance concerns, recent studies have utilized multilayer perceptron, support vector machine (SVM), and other methods. However, these techniques have limitations and are inefficient when dealing with large data sets, such as system and network data. Considering the use of intrusion detection systems in analyzing extensive traffic data, it becomes necessary to have an efficient classification technique to tackle this issue. This study centers on this issue and employs popular machine learning methods, specifically SVM, random forest, and extreme learning machine (ELM). These techniques are renowned for their classification capabilities. The assessment is carried out utilizing the NSL-knowledge discovery and data mining dataset, renowned as a standard for evaluating intrusion detection mechanisms in the industry. The results indicate that ELM surpasses other approaches in terms of performance.

Introduction

Intrusion poses a significant threat to security, as it can result in data theft, deletion, and hardware damage within a matter of seconds. The repercussions of an intrusion extend beyond immediate consequences, with financial losses and compromised critical IT infrastructure leading to information inferiority in cyber warfare. Consequently, the importance of intrusion detection and prevention cannot be overstated.

Accuracy is a crucial factor in intrusion detection techniques, as it directly impacts the detection rate and false alarm rate. Addressing the issue of accuracy is essential to reduce false alarms and increase the overall effectiveness of intrusion detection. Motivated by this concern, this research work aims to explore the application of support vector machine (SVM),

random forest (RF), and extreme learning machine (ELM) in addressing the classification problem associated with intrusion detection.

To evaluate the performance of intrusion detection mechanisms, a standardized dataset, namely NSL-knowledge discovery and data mining (KDD), is utilized. This dataset is an enhanced version of the KDD dataset and is widely acknowledged as a benchmark for evaluating intrusion detection methods. By employing this dataset, the study aims to provide a comprehensive assessment of the effectiveness of the proposed intrusion detection techniques.

Related Work

Securing computer and network data is highly critical for both organizations and individuals, as compromised information can result in significant harm. In this regard, intrusion detection systems are essential. The development of machine learning techniques has sparked the introduction of different methods designed to improve the efficiency of intrusion detection systems.
Wang et al. [1] presented an intrusion detection framework based on support vector machines (SVM) and evaluated their method using the NSL-KDD dataset. They claimed superiority over other approaches with an effectiveness rate of 99.92%. However, they did not provide detailed information about the dataset statistics or the number of training and testing samples used. Moreover, SVM's performance tends to decrease when analyzing large datasets, making it suboptimal for intrusion detection involving extensive network traffic.

Kuang et al. [2] employed a hybrid model of SVM and kernel principal component analysis (KPCA) with genetic algorithms (GA) for intrusion detection. Their system achieved a detection rate of 96% using the KDD CUP99 dataset. However, this dataset has limitations, such as redundancy, which biases the classifier towards more frequently occurring records. Additionally, KPCA's feature reduction approach may result in the loss of important features, as it selects only the top percentages of principal components from the principal space. Furthermore, SVM is not well-suited for handling heavy data, particularly when monitoring high network bandwidth.

Intrusion detection systems offer assistance in detecting, preventing, and resisting unauthorized access. Aburomman and Reaz [3] proposed an ensemble classifier method combining particle swarm optimization (PSO) and SVM, achieving an accuracy of 92.90%. They utilized the KDD99 dataset, which also suffers from the aforementioned drawbacks. Additionally, SVM's performance degrades as the data size increases, making it less suitable for analyzing large datasets.

Raman et al. [4] introduced an intrusion detection mechanism based on a hypergraph genetic algorithm (HG-GA) for parameter setting and feature selection in SVM. They claimed superiority over existing approaches, achieving a detection rate of 97.14% on the NSL-KDD dataset, commonly used for experimentation and validation in intrusion detection systems.

Given the criticality of network system security in our daily lives, Teng et al. [5] conducted important research. They developed a model based on decision trees (DTs) and SVMs, testing it on the KDD CUP 1999 dataset. The results showed an accuracy of 89.02%. However, due to high computation costs and poor performance, SVMs are not preferred for heavy datasets.

Farnaaz and Jabbar [6] devised an intrusion detection model based on random forest (RF) and evaluated its effectiveness using the NSL-KDD dataset. Their results demonstrated a detection rate of 99.67% compared to J48. Nonetheless, the main limitation of the RF algorithm lies in its potential slowness for real-time prediction due to the formation of numerous trees. Elbasiony et al. [7] proposed an intrusion detection model based on RF and weighted k-means, achieving an accuracy of 98.3% on the KDD99 dataset. Similar to RF, the slowness of real-time prediction is a drawback, attributable to the large number of trees formed. Additionally, the KDD99 dataset exhibits limitations as previously discussed.

Method:

The proposed model encompasses several key phases, namely the dataset, pre-processing, classification, and result evaluation. Each phase plays a crucial role and significantly contributes to the overall performance of the system. This work specifically aims to explore and compare the performance of three classifiers, namely SVM, RF, and ELM, in the context of intrusion detection. Figure 1 illustrates the proposed model for the intrusion detection system in this study.

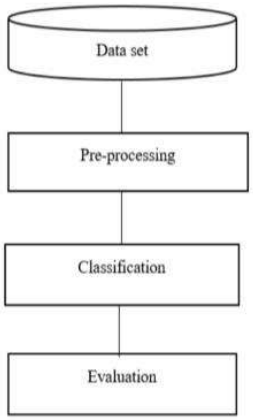


Figure 1: Proposed model of intrusion detection system.

A. Dataset

Selecting an appropriate dataset for experimentation is a critical task, as the performance of the system relies on the accuracy of the dataset. The more accurate the data, the higher the effectiveness of the system. There are various methods to obtain a dataset, including 1) sanitized dataset, 2) simulated dataset, 3) testbed dataset, and 4) standard dataset [8]. However, challenges arise when applying the first three methodologies. The real traffic method can be costly, while the sanitized method may compromise safety. Developing a simulation system is complex and presents its own challenges. Additionally, modeling various network attacks requires different types of traffic, which can be intricate and expensive. To overcome these difficulties, the NSL-KDD dataset is utilized in this study to validate the proposed intrusion detection system.

B. Pre-Processing

Pre-processing is necessary to eliminate or replace these features as they do not significantly contribute to intrusion detection. However, this pre-processing step introduces certain overhead, such as increased training time and added complexity to the classifier's architecture, which can result in wasted memory and computing resources. Therefore, to enhance the performance of intrusion detection systems, non-numeric features are excluded from the raw dataset. This exclusion helps streamline the process and optimize the system's overall efficiency.

C. Classification

Intrusion detection systems play a crucial role in distinguishing between normal and intrusive activities through the use of an intrusive analysis engine. The literature reports the use of different classifiers, such as multilayer perceptron, support vector machine (SVM), naive Bayes, self-organizing map, and decision tree (DT), as intrusive analysis engines in intrusion detection.

In this study, however, three specific classifiers, namely SVM, random forest (RF), and extreme learning machine (ELM), are employed due to their well-established capabilities in handling classification problems. Each classification approach is further described in detail below.

1) Support Vector Machine

Support vector machines (SVMs) were initially proposed by Vapnik (1995) as a solution for classification and regression analysis [9]. SVM is a supervised learning technique that effectively classifies data from various domains into different categories. It is commonly employed in two-class classification problems and can handle both linear and non-linear data classification tasks. SVM constructs one or multiple hyperplanes in a high-dimensional space, with the optimal hyperplane being the one that best separates the data into distinct classes with the maximum margin between them. To handle non-linear classification, SVM utilizes various kernel functions, including linear, polynomial, radial basis, and sigmoid, to estimate the margins. These kernel functions aim to maximize the separation between hyperplanes. SVMs have gained significant attention from researchers due to their promising applications in fields such as image processing and pattern recognition [10].

Figure 2 presents the architecture of the SVM classification model used in the proposed intrusion detection system. The implementation of the SVM model in our system utilizes the radial basis function (RBF) kernel. The RBF kernel calculates the squared Euclidean distance between two numeric vectors and maps the input data to a high-dimensional space, enabling an optimal separation of the data into their respective attack classes. The RBF kernel is particularly effective in handling data sets with complex boundaries. In our study, we employed the freely available LibSVM package [11] for conducting all simulations. Since our problem involves multiclass classification, the one-vs-all approach is employed for attack classification. This approach divides the multiclass problem into a series of two-class problems. The RBF kernel is utilized in this study, and its mathematical representation is as follows:

$$K(x,y) = e^{-\gamma\|x-y\|^2}, \quad \gamma > 0 \qquad (1)$$

For given training samples (xi,yi) , i=1,2,...n , where i is the maximum number of samples in the training data, xi∈Rn and yi∈{1,−1} , where 1 shows samples from a positive class and −1

represents sequences from the negative class. When using SVM, the solution of the following problem is provided.

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to } y_i \left(w^T w \phi \left(x_i \right) + b \right) \geq 1 - \xi_i.$$

Here, ϕ transforms the training vector xi to the higher dimensional space. Following this, the SVM shows a hyper-plane having a maximum margin to separate different classes of data.

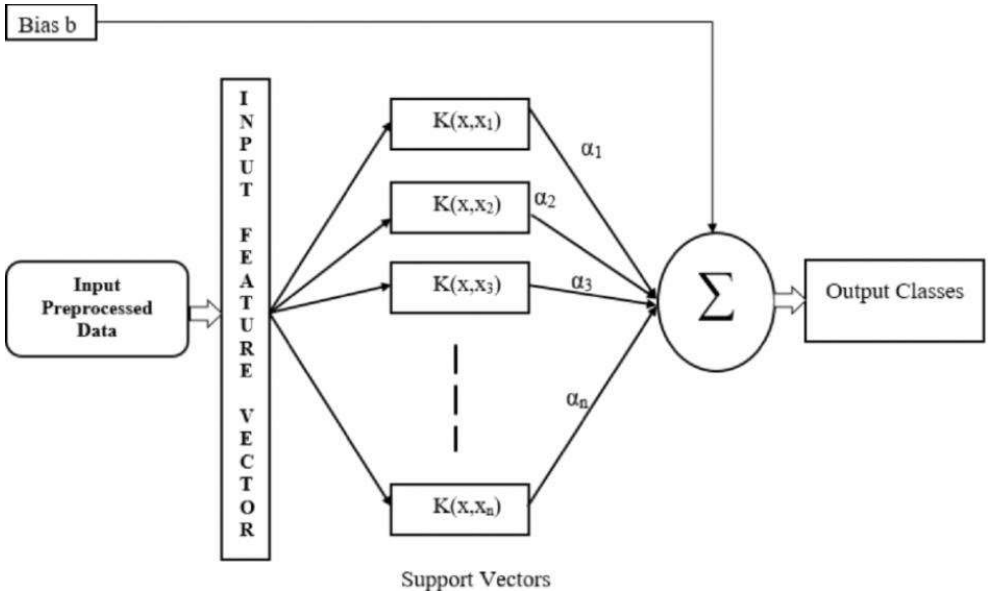


FIGURE 2. Architecture of SVM for intrusion detection.

The observed results obtained through the SVM model are not notably convincing in comparison to the outcomes produced by other classifiers. One advantage of SVM is its ability to perform well with minimal parameter adjustment. However, there are certain drawbacks associated with SVM. For instance, it requires the use of a Gaussian function for each training set instance, resulting in increased training time. Additionally, SVM's performance may degrade when handling very large datasets consisting of thousands of instances, such as in the case of classification tasks. In situations where the maximum margin classifier fails to find a separating hyperplane, a soft margin approach can be employed to address this issue. The soft margin technique incorporates positive slack variables

ξi, where i = 1, 2, ..., N, in the constraints, which can be expressed as follows:

$$(w.\ x_i - b) \geq +1 - \xi_i \quad \text{for } y_i = +1$$
$$(w.\ x_i - b) \geq -1 + \xi_i \quad \text{for } y_i = -1$$
$$\xi \geq 0.$$

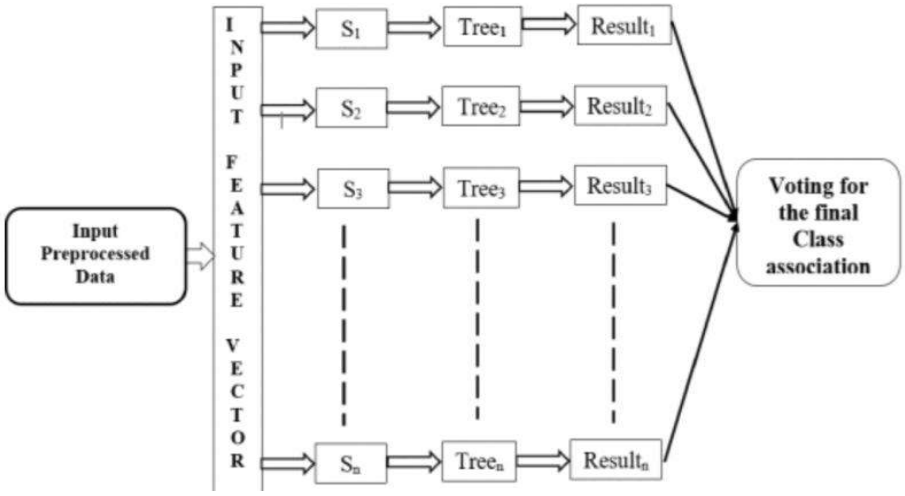
When an error occurs, ξi must exceed unity. Then, ∑iξi is an upper bound on the training error. The Lagrange in this situation is as follows:

$$L_p = \frac{1}{2}\|w^2\| + C\sum_{i=1}^{n}\xi_i - \sum_i \alpha_i \{y_i (x_i . w - b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i,$$

2) Random Forest

Random forests (RFs) are ensemble classifiers commonly used for classification and regression analysis in intrusion detection data. The RF approach involves generating multiple decision trees during the training phase and obtaining class labels through majority voting [12]. RFs exhibit high classification accuracy and are capable of handling outliers and noise in the data. The decision to utilize RF in this study is based on its resilience to overfitting and its proven track record of delivering good classification results.

Figure 3 showcases the implementation of the random forest classification model in the data classification process within the proposed system. A pre-processed sample consisting of n instances is inputted into the random forest classifier. RF creates n different trees using various feature subsets. Each tree produces its own classification result, and the final classification is determined by majority voting. The sample is assigned to the class with the highest voting scores. Previous classification results indicate that RF is well-suited for classifying such data, often yielding superior performance compared to other classifiers. Additionally, RF offers advantages such as higher accuracy when compared to Adaboost and a reduced risk of overfitting.



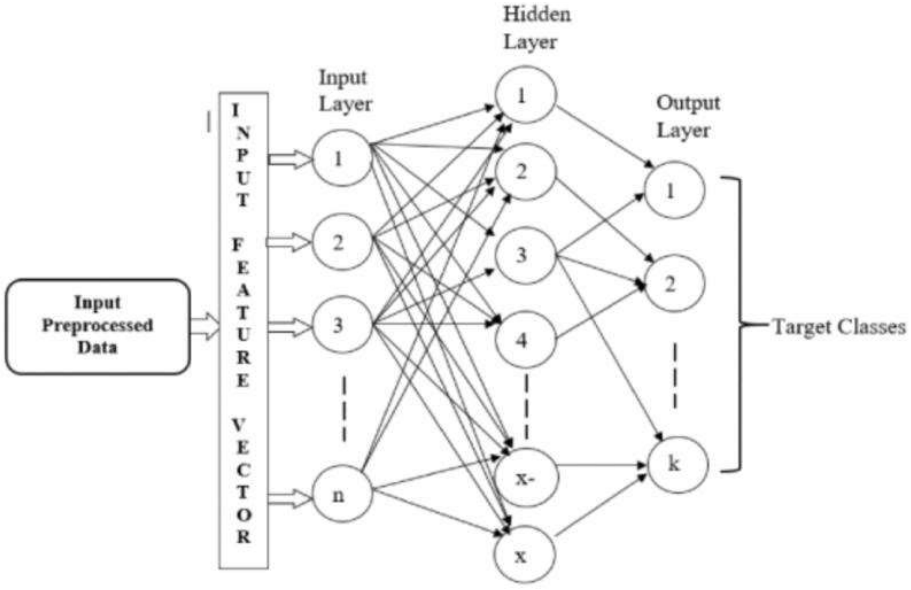
3) Extreme Learning Machine

Extreme learning machine (ELM) refers to single or multiple hidden layer feedforward neural networks [13]. ELM serves as a versatile tool for tackling a range of problems, including classification, clustering, regression, and feature engineering.

The learning algorithm employed by ELM consists of an input layer, one or multiple hidden layers, and an output layer. In traditional neural networks, the adjustment of input and hidden layer weights is computationally expensive and time-consuming, requiring multiple iterations to converge.

To address this challenge, Huang et al. [13] proposed a Single Layer Feedforward Network (SLFN), which involves the arbitrary selection of input weights and hidden layer biases to minimize training time. Comprehensive details on ELM can be found in the works of Huang et al. [14] and Qayyum et al. [15].

The authors assert that these models exhibit faster learning capabilities and achieve superior generalization compared to other feedforward network models. ELM demonstrates comparable performance to SVM and other state-of-the-art machine learning classifiers, while particularly excelling in handling highly complex datasets. The architecture of the proposed system is illustrated in Figure 4.



In the context of intrusion detection, we have N input samples represented as (zi, yi), where zi = [xi1, xi2, ..., xin]T represents the ith sample with n different features, and yi = [yi1, yi2, ..., yim]T denotes the actual labels of xi. A traditional Single Layer Feedforward Network (SLFN) with K hidden neurons can be defined as follows:

$$\sum_{m=1}^{K} \beta_i h\left(w_m . x_i + c_m\right) = \alpha_i, \quad i = 1, \ldots\ldots\ldots, N$$

Here, wm = [wm1, wm2, ..., wmn]T represents the chosen weight vector, indicating the connection of the ith hidden neuron with the input nodes. The weight vector, βi = [βi1, βi2, ..., βim]T, signifies the connection between the ith hidden neuron and the output nodes, while cm denotes the threshold of the ith hidden neuron. Similarly, αk = [αk1, αk2, ..., αkm]T represents the kth output neuron. The activation function h(.) is applied to each hidden neuron. The SLFN, with M hidden neurons and an appropriate activation function, can approximate the N training samples with zero error.

Apart from SLFN, different techniques have been utilized for identifying and categorizing intrusions in wired and wireless environments [16]-[20].

D. Evaluation

The assessment of the developed system is performed using the NSL-KDD benchmark dataset. The dataset is divided randomly into three

subsets: the complete dataset, the half dataset, and the 25% dataset. The complete dataset contains 65,535 samples, the half dataset has 32,767 samples, and the 25% dataset consists of 18,383 samples. Accuracy, precision, and recall are employed as evaluation criteria to gauge the system's performance. These criteria are widely used to evaluate the efficiency of intrusion detection classification algorithms [21].

Accuracy: Accuracy is computed as "the total number of correct predictions, True Positive (TP) + True Negative (TN) divided by the total number of a dataset Positive (P) + Negative (N)".

$$Accuracy = \frac{TP + TN}{P + N}$$

Precision: Precision is determined by dividing the count of accurate positive predictions (TP) by the sum of positive predictions (TP + FP).

$$Precision = \frac{TP}{TP + FP}$$

Recall: The calculation for recall involves dividing the total number of correct positive predictions (TP) by the total number of positives (P). Recall is also known as the true positive rate or sensitivity.

$$Recall = \frac{TP}{P}$$

Results:

Figure 5 illustrates the accuracy of SVM (Linear), SVM (RBF), RF, and ELM on 20% testing and 80% training data samples. ELM demonstrates superior accuracy compared to SVM (Linear), SVM (RBF), and RF on the full dataset. However, SVM (RBF) shows improved accuracy over RF and ELM on half of the dataset. In terms of accuracy on 1/4th of the dataset, SVM (Linear) outperforms the other techniques, as depicted in Figure 5.

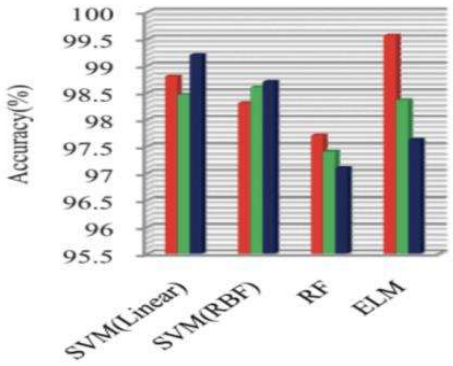


Figure 6 illustrates the precision of SVM (Linear), SVM (RBF), RF, and ELM on a split of 20% testing data and 80% training data samples. ELM exhibits better precision than SVM (Linear) and SVM (RBF) on the full dataset, outperforming RF as well. On half of the dataset, SVM (Linear) demonstrates higher precision compared to SVM (RBF), ELM, and RF. On 1/4th of the dataset, the precision of SVM (Linear) is equal to that of SVM (RBF). Moreover, in the 25% dataset, SVM exhibits better performance compared to ELM and RF.

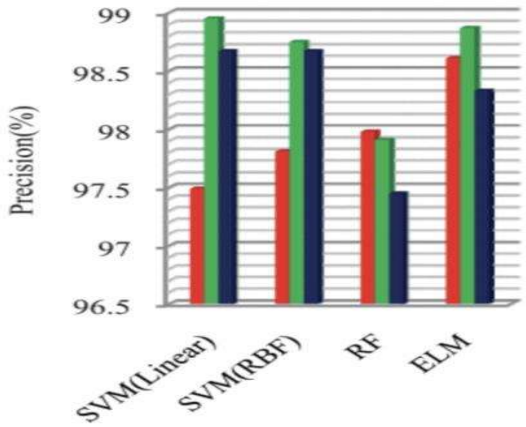


Figure 7 illustrates the recall rates of SVM (Linear), SVM (RBF), RF, and ELM when tested on 20% of the data and trained on 80% of the data samples. ELM achieves better recall than SVM (Linear), SVM (RBF), and RF on the full dataset. SVM (Linear) exhibits greater recall compared to SVM (RBF), ELM, and RF. On 1/4th of the dataset, the ranking of recall is as follows: SVM (RBF) is first, SVM (Linear) is second, RF is third, and ELM is fourth. These findings suggest that SVM excels with smaller datasets, whereas ELM surpasses other methods with larger datasets.

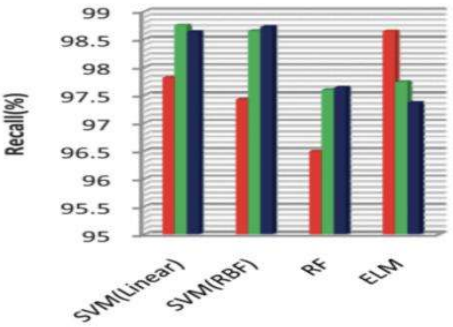


Figure 9 showcases the precision of SVM (Linear), SVM (RBF), RF, and ELM on 10% testing and 90% training data samples. ELM exhibits better precision than RF, SVM (RBF), and SVM (Linear) on the full dataset. SVM (Linear) demonstrates higher precision on half of the dataset compared to

ELM and RF. Additionally, SVM (Linear) performs better than ELM and RF on 1/4th of the dataset.

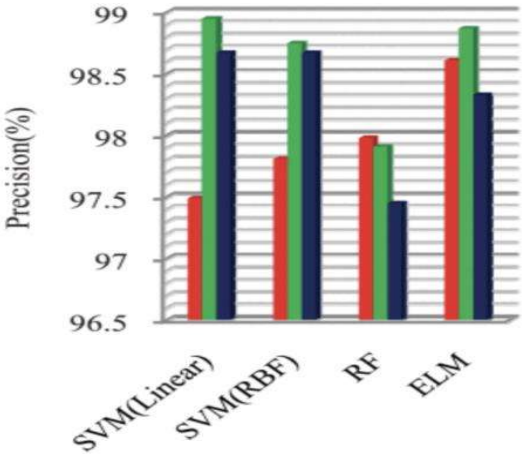
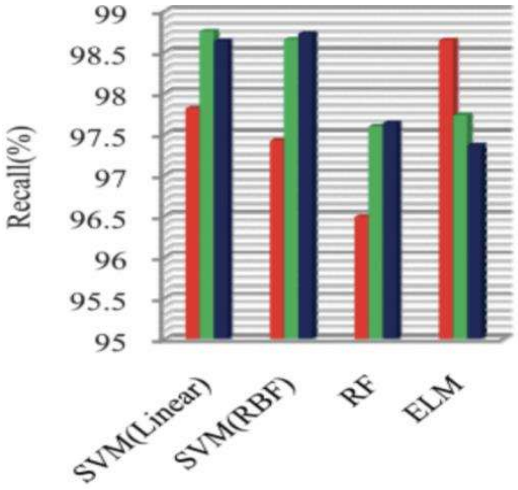


Figure 10 displays the recall of SVM (Linear), SVM (RBF), RF, and ELM on 10% testing and 90% training data samples. ELM outperforms SVM (Linear), SVM (RBF), and RF on the full dataset in terms of recall. SVM (Linear) exhibits better recall than SVM (RBF), ELM, and RF on half of the dataset. On 1/4th of the dataset, SVM (RBF) is almost equal to SVM (Linear) in terms of recall, while surpassing RF and ELM, as depicted in Figure 10.



Conclusion

Intrusion detection and prevention play a crucial role in the reliability and security of present and future networks and information systems, as our daily activities increasingly rely on them. Moreover, with the adsvent of the Internet of Things, the challenges in this domain are expected to become even more formidable. In this context, intrusion detection systems have gained significant

importance in recent decades. While various techniques have been employed in intrusion detection systems, machine learning approaches have gained prominence in recent literature. Additionally, different machine learning techniques have been utilized, but certain techniques are better suited for analyzing vast amounts of data in the context of network and information system intrusion detection. To tackle this challenge, this work investigates and compares different machine learning techniques, namely SVM, RF, and ELM. Among these approaches, ELM exhibits superior performance in terms of accuracy, precision, and recall on the complete dataset, which consists of 65,535 records comprising both normal and intrusive activities. Furthermore, SVM yields better results than the other techniques on half of the dataset and a quarter of the dataset. As a result, ELM emerges as a suitable technique for intrusion detection systems designed to analyze large volumes of data. In the future, further exploration of ELM is warranted to assess its performance in feature selection and feature transformation techniques, thereby enhancing its capabilities in the field of intrusion detection.

References

[1] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," Knowl.-Based Syst., vol. 136, pp. 130–139, Nov. 2017, doi: 10.1016/j.knosys.2017.09.014.

[2] F. Kuang. S. Zhang and W. Xu, "A novel hybrid SVM and KPCA with the GA model for intrusion detection," Appl. Soft Comput., vol. 18, pp. 178–184, May 2014, doi: 10.1016 j.asoc.2014.01.028.

[3] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," Appl. Soft Comput., vol. 38, pp. 360–372, Jan. 2016, doi: 10.1016/j.asoc.2015.10.011.

[5] S. Teng, N. Wu, H. Zhu, L. Teng, and W. Zhang, "SVM-DT-based adaptive and collaborative intrusion detection," IEEE/CAA J. Automatica Sinica, vol. 5, no. 1, pp. 108–118, Jan. 2018, doi: 10.1109/JAS.2017.7510730.

[6] N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," Proc. Comput. Sci., vol. 89, pp. 213–217, Jan. 2016, doi: 10.1016/j.procs.2016.06.047.

[7] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," Ain Shams Eng. J., vol. 4, no. 4, pp. 753–762, 2013, doi: 10.1016/j.asej.2013.01.003.

[8] I. Ahmad and F. e Amin, "Towards feature subset selection in intrusion detection," in Proc. IEEE 7th Joint Int. Inf. Techno. Artifi. Intellig. Conf., Chongqing, China, Dec. 2014, pp. 68–73.
[9] J. Jha and L. Ragha, "Intrusion detection system using support vector machine," Int. J. Appl. Inf. Syst., vol. ICWAC, no. 3, pp. 25–30, Jun. 2013.
[10] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by timevarying chaos particle swarm optimization," Neurocomputing, vol. 199, pp. 90–102, Jul. 2016.
[11] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 27:1–27:27, 2011.
[12] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random forest," in Information Computing and Applications, B. Liu, M. Ma, and J. Chang, Eds. Berlin, Germany: Springer, 2012, pp. 246–252.
[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in Proc. IEEE Int. Joint Conf. Neural Netw., vol. 2, Jul. 2004, pp. 985–990, doi: 10.1109/IJCNN.2004.1380068.
[14] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 2, pp. 513–529, Apr. 2012, doi: 10.1109/TSMCB.2011.2168604.
[15] A. Qayyum et al., "Image classification based on sparse-coded features using sparse coding technique for aerial imagery: A hybrid dictionary approach," in Neural Computing and Applications. London, U.K.: Springer, 2017 pp. 1–21, doi: 10.1007/s00521-017-3300-5.
[16] A. Derhab, A. Bouras, M. R. Senouci, and M. Imran, "Fortifying intrusion detection systems in dynamic ad hoc and wireless sensor networks," Int. J. Distrib. Sensor Netw., vol. 10, no. 12, p. 608162, 2014.
[17] I. Yaqoob et al., "The rise of ransomware and emerging security challenges in the Internet of Things," Comput. Netw., vol. 129, pp. 444–458, Dec. 2017.
[18] A. R. Sfar, E. Natalizio, Y. Challal, and Z. Chtourou, "A roadmap for security challenges in the Internet of Things," Digit. Commun. Netw., vol. 4, no. 2, pp. 118–137, 2017, doi: 10.1016/j.dcan.2017.04.003.
[19] A. A. Aziz, S. EL-Ola Hanafi, and A. E. Hassanien, "Comparison of classification techniques applied for network intrusion detection

and classification," J. Appl. Log., vol. 24, pp. 109–118, Nov. 2017, doi: 10.1016/j.jal.2016.11.018.
[20] I. Ahmad, "Feature selection using particle swarm optimization in intrusion detection," Int. J. Distrib. Sensor Netw., vol. 11, no. 10, p. 806954, 2015, doi: 10.1155/2015/806954.