Enhancing Data Security in Cloud Computing with Optimized Feature Selection and Machine Learning for Intrusion Detection

S Kavitha
NITTE Meenakshi Institute of Technology
Swetha Gadde
R.VR. & J.C College of Engineering
Ramya Thatikonda
University of the Cumberlands
Srinivas Aditya Vaddadi
University of the Cumberlands
E Naresh (naresh.e@manipal.edu)
Manipal Academy of Higher Education
Piyush Kumar Pareek
NITTE Meenakshi Institute of Technology

Research Article

Keywords: Cloud Computing, Data Collection, Feature Extraction, Machine Learning Models, Accuracy Analysis

Posted Date: November 9th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3572347/v1

Additional Declarations: No competing interests reported.

Abstract

The major objective of this project is to create a Machine Learning (ML) model that can improve data security when data is transported or handled utilizing cloud computing. Researchers must develop a model or a strategy that can secure the data because cloud computing is one of the fastest-growing technologies. This study employs a data set made up of several parameters that can help ML models improve the security of the data for this goal. Then, using a variety of methods, this data set is prepossessed. Following feature extraction, when the parameters are left out, the preprocessed data is then provided to the message learning models. The ML models are then trained and tested using these exclusive parameters. For greater efficiency, three separate learning metrics and three different ML models are created. They are the Artificial Neural Network (ANN) algorithm, the K-nearest neighbor (KNN) algorithm, and the Random Forest (RF) algorithm. The models' testing and training are then used to compare their efficacy. For this reason, the outcomes of both training and testing are noted and examined. The ideal measuring methodology for improving data security is found using parameters like precision, true positive rate, false negative rate, etc. The effectiveness of the feature extraction technique is tabulated alongside the ML models to determine the optimal feature extraction technique and ML model combination. In the end, it is discovered that the ANN algorithm combined with a simulated bee colony may generate the highest level of efficiency. This model's output has a final accuracy of 93.8%. As a result, this approach can be used in conjunction with applications that call for cloud computing, improving data security in the process.

I. Introduction

Cloud computing is a niche technology that allows users to save and access data across the Internet instead of having a physical hardware storage element. The use and management of technology resources by corporations and individuals have been completely changed by cloud computing. A variety of computing resources can be accessed via the internet thanks to the technology and service delivery paradigm known as cloud computing. Users and businesses can use cloud services offered by third-party suppliers in place of owning and maintaining physical infrastructure and software.

These cloud services cover a range of remotely provided and managed resources, including processing power, storage, databases, networking, software, and much more. It makes data accessible and storable from any location having an internet connection. Although there are many advantages to this convenience, if adequate security precautions are not taken, it also increases the risk of unwanted access. Cloud service providers frequently host data and applications for several clients on shared infrastructure. Without strong security measures, there is a chance that data could leak or that another customer's data or application could be compromised.

One such solution is provided. The study analyzes various feature extraction techniques and ML models to find the best combination of the two which can be employed as a part of cloud computing that can

increase the security of the data involved. The workflow and the construction of the ML models are explained elaborately in the upcoming chapters of this paper.

II. Literature Survey

The research [1] has proposed an idea of using necessary parameters that can potentially harm the security of the company a data as a barrier. Because the motivation of the organization's users is necessary for it to function correctly, practical methods to minimize information security stress are required in order to increase information security. Therefore, the goal of this study is to identify the crucial elements that can improve security while lowering employees' stress levels related to information security. As a result, the theory of protection motivation has defined security and confidence pressure in policies as influencing factors for clarifying changes in security reinforcing actions and risk effectiveness, which is and responses expenses associated with cyberattacks are taken into account as requirements. This research provides a solution to the issue of security reinforcement by investigating the elements that influence organizational member behavior and could raise members' motivation for defense. The study [2] used cryptography as a form of data security. The divide in resource sharing has been undermined by cloud connectivity. The security risk increases with the ease with which data can be accessed from the cloud. Security is a crucial component in lowering security risk. Technology should be trusted, but control is preferable because data owners are not always present. Before moving data into a public environment, a cryptographic mechanism called identity-based proxy re-encryption is employed to convert the data. Since the owner of the data is not present, a rule-based data-centric solution is suggested to maintain data security. Role-based authorization access control is created to restrict illegal access to data. The remedy is created with OpenStack.

The research [3] analyzed various datasets and used IoT as a main element of their data security module. This study offers a network threat detection system that uses Artificial Intelligence (AI) to perform data screening, preliminary processing, and data learning for network flow and cybersecurity event data in order to identify unknown and emerging 5G network incursions. First, the proposed system's performance is assessed using two famous datasets—NSL-KDD and CICIDS—before being put to the test in real-world 5G Industrial Internet of Things (IIoT) settings. This study makes use of the model, which is reduced to a genuine smart factory that includes a number of industrial devices based on IIoT, to demonstrate invasion against virtual threats in real scenarios. The study [4] used IoT in data security. Data transfer between smart things can be accomplished by using sensing networks, networking, and computing functions as IT develops and IPv6 technology finally matures. The IoT is the name given to this idea as it develops in a network context. The challenges surrounding data security have not, however, been covered in the linked IoT research. In order to safeguard data in the perception layer and reduce excessive computational burdens, this study constructs security-level agreements. It then calculates the computational cost of data encryption in the perception layer.

The research [5] developed a novel technique that can predict and stop the man-in-the-middle. NDN, which is a form of content-centric networking encrypts every data packet with a custom paraphrase in

order to confirm its integrity and validity. Comparing this plan to IP networking, many of the security problems can be resolved naturally. Since NDN conceals the specifics of the point-to-point connection, it also supports mobility. However, when a user first connects to a network, a severe attack occurs. In order to fabricate relevant data packets authorized with its own private key, a malicious Man-in-the-middle node can stop the user and keep interjecting the packets. NDN users are severely exposed to security and privacy risks since they cannot perceive attacks without knowledge of and faith in the network. Container Visualization is proposed as a data security mechanism by researchers [6]. Weak isolation is used in container cloud adoption of container virtualization technology. The strategy described in this research for improving container cloud security is based on multiple security domains and offers a comprehensive resource isolation solution. The journal [7] used a combination of Obfuscation and steganography for data protection. This study suggests a sophisticated and original way to increase data security by combining steganography and obfuscation techniques. Obfuscation and steganography are combined in the suggested confidentiality method. The primary concept of steganography is to conceal the availability of information, whereas the essential concept of obfuscation is converting data into a substitute form while masking the original data. The obfuscated text can be obscured inside a picture by using the Least Significant Bit (LSB) replacement technique. The studies' findings demonstrate that the recommended method has a high capacity for incorporation and generates images of excellent quality.

III. Materials and Methods

A dataset named NSL-KDD dataset is collected from the internet. This dataset is the one that is used to train and test the designed ML models. The process in which this is achieved is pictorially represented in Fig. 1.

From Fig. 1, it is clear that the preprocessing is the first step of this research. After preprocessing the images are sent to the feature extraction techniques. The feature extraction techniques employed in this study are Artificial Bee Colony (ABC) and Harmony Search Optimization (HSO). After the featured selection the images are then sent to the ML models. Initially, the models are trained using the selected features. Once the models are properly trained, the efficiency of the models is then instead using a part of the data set that was not used during the training. The results of the training and the testing are tabulated and analyzed to find the best algorithm that can be employed for data security enhancement.

IV. Data collection and Preprocessing techniques

A dataset named NSL-KDD is a collection of data correlated by the University of New Brunswick, Canada. This dataset is an updated version of the KDD' 99 dataset which was developed by the same university. The sole purpose of this data set is to assist researchers by comparing different detection methodologies. This data set is collected and is used as training material in this study. A sample of this data set is shown in Table 1.

Table 1
Sample dataset

Data	Normal	DoS	Probe	U2R	R2L
TRAIN	67343	45927	11656	52	995
TEST	9711	7456	2421	200	2756

The parameters shown in this data set are then used to train the models to enhance the security of the data when cloud computing is involved. But before using this dataset it has to be preprocessed using certain techniques to provide maximum efficiency. The preprocessing techniques used in this sparse feature merging, string feature digitization, and numerical normalization. These techniques are explained further in this chapter.

A. Sparse Feature Merging (SFM)

To combine or merge several related features into a single, more useful feature, smart feature merging is a technique used in feature engineering and ML [8]. The dataset's dimensionality will be decreased, extraneous data will be eliminated, and adding more useful input features could help the model perform better. Smart feature merging has a variety of advantages, including a decrease in overfitting, an increase in model interpretability, and the possibility to shorten training and prediction durations by utilizing fewer features. Integrating features can potentially result in information loss if done incorrectly, therefore it should be done cautiously.

B. String Feature Digitization (SFD)

In data preparation and ML, the process of "string feature digitization," also known as "string encoding" or "categorical feature encoding," involves converting categorical or textual data that can be expressed as strings into a numerical representation [9]. Several ML algorithms need numerical inputs, and these algorithms may find it difficult or impossible to operate on strings directly. This translation is therefore required. When choosing an encoding technique, it is crucial to take into account elements such as cardinality (number of unique categories), the possibility of overfitting, and the interpretability of the final numerical representation.

C. Numerical Normalization (NN)

In the context of data preprocessing, the act of normalizing numerical attributes in a dataset to a consistent spectrum or distribution is referred to as numerical normalization. Making sure that distinct features have comparable scales is the aim of normalization, which might be crucial for many ML techniques. Normalization improves the data's suitability for modeling and analysis without altering the underlying data relationships [10]. The performance of ML models can be enhanced by numerical normalization, particularly those that use distance-based calculations (like k-nearest neighbors) or optimization methods (like gradient descent). The analysis or modeling process is prevented from being

dominated by features with higher ranges or variances by putting the data into a similar scale or distribution.

V. Feature Extraction

In the context of improving data security, the process of choosing, converting, or displaying pertinent information using raw data in a way that makes it useful for security-related tasks is referred to as feature extraction. In terms of data security, it is essential for identifying breaches and recognizing anomalies, authorization, and other processes. Various tools and methods can be used to perform feature extraction. In this study, 2 different techniques named the ABC and the harmony search of the optimization are used for future extraction.

A. Artificial bee colony

The ABC algorithm is an optimization technique that draws inspiration from nature and is used to address a variety of computing issues, particularly optimization issues. Dervis Karaboga first proposed it in 2005, drawing inspiration from the foraging habits of honeybee colonies. The flow of execution is pictorially represented in figure 2.

The ABC algorithm simulates how bees find food sources and communicate with one another inside a colony [11]. The success of ABC, however, is dependent on the problem being addressed and the proper parameter setup, much like with any optimization technique. The ABC algorithm has undergone several modifications and adaptations to address certain issues and enhance its functionality in a variety of applications.

There is a population of manufactured bees (solutions). Up, until a conclusion requirement, such as a particular amount of iterations or the discovery of a suitable resolution, is fulfilled, the algorithm iterates through the phases. A food source's possibility of being picked is inversely correlated with its quality; better options are more likely to be selected. It is flexible and can be used to solve a variety of optimization issues, such as perpetual, discrete, and sequential issues. The expertise about the problem and parameter tuning is low.

B. Harmony search optimization

The nature of the problem being solved and the parameter settings of the Harmony Search algorithm might affect how well it performs [12]. It is usual practice to enhance the algorithm's performance by tweaking these parameters and tailoring the algorithm to particular problem areas. Engineering, scheduling, and parameter optimization in ML are only a few of the many applications it has been used. The Harmony Search (HS) optimization method mimics the technique performed by musicians in an orchestra to find a harmonic melody in order to identify the best solution to a problem. In 2001, Zong Woo Geem created this algorithm, which has since been used to solve numerous optimization issues. The flow chart representing the flow of HSO is shown in Figure 3.

There is a starting population of "harmonies" produced. Every harmony stands for a possible answer to the optimization issue. Within the boundaries of the task, these harmonies are generated at random. A fitness function assesses each harmony's quality. The effectiveness of harmony in resolving the issue is measured by this function. The Harmony Memory (HM), is a memory kept by the algorithm of the best harmonies discovered thus far. The harmony with the best fitness values is stored in HM. Up until a termination requirement is fulfilled, the algorithm iterates through harmonic improvisation and memory update. The optimal harmony is then recovered from the Harmony Memory as the final step in solving the optimization problem.

VI. ML Algorithms

By assisting enterprises in identifying, averting, and addressing security threats and vulnerabilities, ML algorithms serve a vital part in improving data security. To enhance security measures, they make use of data patterns and insights. Large and diverse datasets are best for training ML algorithms, which frequently need ongoing updates to respond to new threats. Despite the fact that these algorithms have the potential to greatly improve data security, they should be a part of a larger security strategy that also includes preventative measures like access restriction, encryption, and routine security audits. Three different ML models are used in this study. In the following section, those algorithms are explained in detail.

A. RF

For prediction tasks, the RF technique is an effective machine-learning tool. The Leo Breiman-created RF algorithm builds individual classification or regression trees for prediction by utilizing bootstrap aggregation (bagging) and random feature selection [13]. In several studies in a variety of fields, including economic forecasting, satellite imaging, genetic and biological analyses, and classification and regression difficulties, RFs have shown to have outstanding predictive powers. RF classifiers are gaining popularity in the field of computer vision, including well-known variations such as Random Ferns along with exceedingly random trees. Ongoing research in the field of RF allows researchers to enhance accuracy, reduce learning and classification time, or achieve both objectives simultaneously. This study focuses on improving RF's precision because it is one of the most effective categorization techniques [14].

Nevertheless, an RF may use less-than-ideal tree classifiers, producing inaccurate classification results, because of the multitude of data distributions in feature spaces with large dimensions. When a significant proportion of poor-quality trees is present in the RF, the collective decision-making of all the trees may result in erroneous decisions. To mitigate this, the research seeks to optimize the RF by identifying and excluding underperforming trees to minimize their detrimental impact on overall performance. Additionally, randomization in RF can lead to correlated trees, potentially impacting performance. The RF's classification accuracy can be increased by reducing the correlation between

trees. This research selects only uncorrelated trees with good classification accuracy in order to improve an extensive amount of decision trees throughout an RF.

An ensemble classification technique called RF integrates the findings of various decision trees. Numerous approaches to creating RFs have been put out over the past ten years, with Breiman's method rising to prominence due to its greater performance over alternatives. The process of building an RF consists of three steps:

- The first phase is Training Data Sampling - Using the bagging approach, randomly sample D with replacement to produce K subgroups of the training data $\{D_1, D_2, \ldots, D_K\}$.
- The second phase is Feature Subspace Sampling and Tree Classifier Building - Given every training session dataset $D_i(1 \leq i \leq K)$, a tree should be grown using a decision tree technique. The optimal split should be chosen as the dividing feature to generate a child node after evaluating all potential splits within each node's subspace X_i (where $F << M$). A tree $h_i(D_i, X_i)$ constructed from training data $D_i$under subspace X_i is the end product of this procedure, which continues until the halting requirements are satisfied.
- The third phase is judgment Aggregation - Create an ensemble classification judgment by using votes from the majority among the K trees $\{h_1(D_1, X_1), h_2(D_2, X_2) \ldots h_K(D_K, X_K)\}$that together make up an RF.

The procedure is mostly driven by the quantity of K trees required to form an RF and the total quantity of F randomly selected features required to build a decision tree. Breiman states that parameter F is calculated as $F = [log_2 M + 1]$ and parameter $K$is commonly set to 100. Use greater values of K and F for datasets that are huge and have a high degree of dimension.

B. KNN

A straightforward and popular supervised ML technique known as K-Nearest Neighbors. It can be used for both classification and regression problems. KNN is a non-parametric, instance-based learning algorithm that generates predictions based on similarities between data points rather than on assumptions about the distribution of the underlying data [15]. KNN is a flexible algorithm that excels at being clear to understand and simple to use. When the decision limit is not very complex, it can be effective in solving difficulties. Ties in classification can happen when different classes receive the same amount of KNN votes. In these circumstances, a number of tie-breaking techniques can be applied. Within high-dimensional spaces or when dealing with unbalanced datasets, it might not function at its best. For successful KNN implementation, hyperparameter adjustment and data pretreatment are frequently required.

To train a KNN model, the following procedures are done. Identify the value of K, which denotes the quantity of closest neighbors to be taken into account while making forecasts. To determine the best value for K, you can experiment with various values. Odd integers like 1, 3, 5, etc. are frequent values for K. Select a distance metric that accurately captures how similar different data points are. Every particular

situation and the properties of the data will determine whatever metric one uses. The K-nearest neighbors' class labels should be determined. Assign the projected class for the new data point to the class label that appears the most frequently among the K neighbors. Try out various K values and different distance metrics to see which one produces the greatest performance for the model on an evaluation or validation set.

C. ANN

ANNs represent a fascinating field of computational models that draw inspiration from the intricate structure of the human brain [16]. They are designed to emulate various aspects of human-like behavior, encompassing critical processes such as learning, adaptation, association, generalization, and abstraction. These functionalities are particularly prominent during the training phase, where ANNs evolve and refine their internal representations based on data. At the core of ANNs are artificial neurons, which serve as the fundamental processing units. These neurons are interconnected in intricate ways, forming a network. One of the remarkable features of ANNs is their remarkable ability to learn from data that is incomplete and laden with noise. This is in stark contrast to traditional computing systems, where a malfunctioning component can lead to a catastrophic system failure. In ANNs, classification is an inherent property thanks to their distributed processing nature. If an individual neuron malfunctions, its erroneous output can be overwritten or compensated for by the correct outputs generated by its neighboring neurons.

The versatility of ANNs makes them a powerful tool for solving complex real-world problems where the relationships between input attributes and desired outputs may not be well understood. They excel in scenarios involving continuous value inputs and outputs, a characteristic that sets them apart from many other ML algorithms. ANNs have been successfully applied in various domains, including handwritten character recognition and medical diagnosis, showcasing their adaptability and efficacy. Moreover, techniques for parallelization can be employed to expedite the computational processes, and recent developments in rule extraction from trained ANNs enhance their utility in data mining tasks, especially in numerical classification and prediction.

Within the realm of ANNs, learning is the central process. It involves iteratively adjusting the synaptic weights that connect artificial neurons to minimize errors. Learning in ANNs is akin to the continuous adaptation of the network's parameters based on environmental stimuli. The type of learning employed depends on how these parameter adjustments are carried out. Two prominent categories are supervised learning, where known input-output pairs $(x_i, y_i)$are provided, and unsupervised learning, where desired output values (y_i) are absent, and the network must uncover patterns and structures in the data independently. Figure 4 illustrates the depiction of the fundamental components of an artificial neuron.

One of the most widely used training algorithms for ANNs, especially in the context of multi-layer perceptrons (ANN-MLP), is the error backpropagation method. This method involves presenting a pattern to the input layer of the network and then processing it layer by layer until the network produces the final

response (f_{mlp}). This response is calculated by considering a combination of synaptic weights ($v_I$and w_{ij}), biases ($b_{I0}$and b_0), and an activation function (ϕ) , as outlined in Eq. (1).

$$f_{mlp} = \phi\left[\sum_1^{N_{on}} v_1 \phi\left(\sum w_{ij} x_l + b_{l0}\right) + b_0\right] \text{ [1]}$$

The MLP training process is significantly influenced by the choice of the learning rate parameter. When the learning rate is set too low, the training of the ANN becomes sluggish, whereas an excessively high learning rate can result in training oscillations, hindering the convergence of the learning process. Typically, this parameter's values fall within the range of 0.1 to 1.0. Training an MLP using the backpropagation algorithm often demands numerous iterations through the training dataset, leading to lengthy training times. If the training process encounters a local minimum, it may struggle to reduce the error for the training set, plateauing at an unacceptable level. One effective strategy for accelerating the learning rate without inducing oscillations is to introduce a momentum term. This constant factor influences how past weight changes affect the current direction of weight adjustments in the weight space. It is advisable to set the momentum rate within the range of 0 to 1 [17].

VII. Result And Discussion

A data set consisting of various parameters regarding the security of the data is obtained from the internet. This data set is then preprocessed and the necessary features are extracted from the data using two different techniques. The extracted features are then sent to the ML models. Three different ML models are designed in this research. They are RF, KNN, and ANN algorithms. The performance of the models during training is tabulated and recorded. This data is then used to compare the efficiency between both models. The performance of the models when the HSO technique is used is shown in Table 2. From Table 2, it can be seen that the ANN algorithm has the highest accuracy of all the other models. It also has the least false positive and false negative rates. The second best is their RF algorithm. For a clearer understanding, the data is also plotted as a bar graph and is shown in Fig. 5.

Table 2
HSO and ML models

Model	RF	KNN	ANN
ACCURACY	96.2337	95.58158	97.29838
PPV	95.85602	95.22277	97.53995
NPV	96.64504	95.97615	97.02578
TPR	96.88656	96.29947	97.36886
TNR	95.53825	94.81045	97.21854
F1	96.36854	95.75809	97.45433
FNR	3.113443	3.700531	2.631139
FPR	4.46175	5.189547	2.781457

The graphic makes it more obvious that of all the algorithms, the one using ANN has the highest accuracy. During an analysis, the false rates are as important as the true positive rates and accuracy. Thus the false positive rates and the false negative rates are plotted on a different graph and shown in Fig. 6.

From the above image, it can be seen that the ANN algorithm has the least false rates stating that it is better than the other two algorithms. Just like HSO, the results are tabulated for ABC as well. Table 3 Consists of the collected data about the performance of the model when the features are extracted using ABC. Even with the ABC technique the ANN algorithm has the best accuracy and higher true rates. The bar graph of the table is shown in Fig. 7.

Table 3
ABC and ML models

Model	RF	KNN	ANN
ACCURACY	98.08358	95.96753	98.38967
PPV	98.29669	95.77309	98.61716
NPV	97.8445	96.17953	98.13384
TPR	98.08272	96.47049	98.34517
TNR	98.08454	95.42736	98.44001
F1	98.18959	96.12052	98.48098
FNR	1.917281	3.529513	1.654827
FPR	1.915456	4.572638	1.559989

The ANN algorithm beats the other two algorithms with a meager difference in all the parameters. Thus the false rates of this feature extraction technique are also plotted as a graph as shown in Fig. 8.

It is clear from the graphic above that the ANN algorithm has the lowest false positive rates when claiming superiority over the other two algorithms. Also when both the feature extraction techniques are compared, it can be seen that the ABC technique has the best results.

VIII. Conclusion

The main aim of this study is to develop an ML model which is capable of enhancing the security of the data when the data is transferred or manipulated using cloud computing. As cloud computing is one of the rapidly advancing technologies it is necessary for the researchers to come up with a model or a design that can secure the data. For this purpose, this study uses a data set consisting of various parameters that can aid the ML models to increase the security of the data. This data set is then prepossessed using various techniques. The preprocessed data is then sent for feature extraction where the parameters are excluded and sent to the message learning models. These exclusive parameters are then used to train and test the ML models. The three different ML models are developed for increasing efficiency. They are the RF algorithm, the KNN algorithm, and the ANN algorithm. The efficiency of the models is then compared by the testing and training of the models. The results of both testing and training are recorded and analyzed for this purpose. Parameters like accuracy, true positive rate and false negative rate etcetera are used to find the ultimate measurement model for data security enhancement. Along with the ML models, the efficiency of the feature extraction technique is also tabulated to find the best combination of the feature extraction technique and the ML model. In the end, it is found that the combination of an ABC and the ANN algorithm can produce the best efficiency of all the other combinations. The final accuracy produced by this model is 93.8%. Thus this model can be deployed along with the application that requires cloud computing which in return provides an enhanced data security.

Declarations

COMPLIANCE WITH ETHICAL STANDARDS

This proposed research has no conflicts of interest with authors.

FUNDING

Authors did not receive any funding for this research.

ETHICAL APPROVAL

This article does not contain any studies with human participants and animals performed by any of the authors.

Research Data Policy and Data Availability

NSL-KDD datasets are collected from the internet, no datasets were created.

References

1. H. Choi and K. J. Young, "Practical Approach of Security Enhancement Method based on the Protection Motivation Theory," 2021 21st ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter), Ho Chi Minh City, Vietnam, 2021, pp. 96-97, doi: 10.1109/SNPDWinter52325.2021.00028.
2. Dixit, R., Ravindranath, K. (2022). Enhancement in Security for Intercloud Scenario with the Help of Role-Based Access Control Model. In: Senjyu, T., Mahalle, P., Perumal, T., Joshi, A. (eds) IOT with Smart Systems. Smart Innovation, Systems and Technologies, vol 251. Springer, Singapore. https://doi.org/10.1007/978-981-16-3945-6_27
3. J. Lee, H. Kim, C. Park, Y. Kim and J. -G. Park, "AI-based Network Security Enhancement for 5G Industrial Internet of Things Environments," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 971-975, doi: 10.1109/ICTC55196.2022.9952490.
4. Wang, SC., Lin, YJ., Yan, KQ., Chen, CW. (2019). Security Enhancement of Internet of Things Using Service Level Agreements and Lightweight Security. In: Arai, K., Kapoor, S., Bhatia, R. (eds) Advances in Information and Communication Networks. FICC 2018. Advances in Intelligent Systems and Computing, vol 887. Springer, Cham. https://doi.org/10.1007/978-3-030-03405-4_15
5. Z. Song and P. Kar, "Name-Signature Lookup System: A Security Enhancement to Named Data Networking," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 2020, pp. 1444-1448, doi: 10.1109/TrustCom50675.2020.00194.
6. Zhong, C., Yuan, X. (2020). A Method of Resource Isolation and Security Enhancement of Container Cloud Based on Multiple Security Domains. In: Xu, Z., Parizi, R., Hammoudeh, M., Loyola-González, O. (eds) Cyber Security Intelligence and Analytics. CSIA 2020. Advances in Intelligent Systems and Computing, vol 1146. Springer, Cham. https://doi.org/10.1007/978-3-030-43306-2_90
7. B. F. Mary and D. I. G. Amalarethinam, "Data Security Enhancement in Public Cloud Storage Using Data Obfuscation and Steganography," 2017 World Congress on Computing and Communication Technologies (WCCCT), Tiruchirappalli, India, 2017, pp. 181-184, doi: 10.1109/WCCCT.2016.52.
8. Hnatushenko, V., Shedlovska, Y., Shedlovsky, I. (2023). Processing Technology of Thematic Identification and Classification of Objects in the Multispectral Remote Sensing Imagery. In: Babichev, S., Lytvynenko, V. (eds) Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making. ISDMCI 2022. Lecture Notes on Data Engineering and Communications Technologies, vol 149. Springer, Cham. https://doi.org/10.1007/978-3-031-16203-9_24

9. Kumar, A., Kumar, N., Prakash, A. (2023). Optimization-Based Speed Control Strategies for Induction Motor Drives in Plug-In Hybrid Electric Vehicle Using Quasi-Opposition Harmony Search Algorithm. In: Namrata, K., Priyadarshi, N., Bansal, R.C., Kumar, J. (eds) Smart Energy and Advancement in Power Technologies. Lecture Notes in Electrical Engineering, vol 926. Springer, Singapore. https://doi.org/10.1007/978-981-19-4971-5_62
10. Phan, T.A., Nguyen, N.D., Thanh, H.L., Bui, KH.N. (2022). Neural Inverse Text Normalization with Numerical Recognition for Low Resource Scenarios. In: Nguyen, N.T., Tran, T.K., Tukayev, U., Hong, TP., Trawiński, B., Szczerbicki, E. (eds) Intelligent Information and Database Systems. ACIIDS 2022. Lecture Notes in Computer Science(), vol 13757. Springer, Cham. https://doi.org/10.1007/978-3-031-21743-2_47
11. Beg, M.S., Waoo, A.A. (2023). Packet Delivery Comparison Using Artificial Bee Colony Algorithm with Dynamic Technique. In: Bhattacharyya, S., Banerjee, J.S., Köppen, M. (eds) Human-Centric Smart Computing. Smart Innovation, Systems and Technologies, vol 316. Springer, Singapore. https://doi.org/10.1007/978-981-19-5403-0_8
12. Kumar, A., Kumar, N., Prakash, A. (2023). Optimization-Based Speed Control Strategies for Induction Motor Drives in Plug-In Hybrid Electric Vehicle Using Quasi-Opposition Harmony Search Algorithm. In: Namrata, K., Priyadarshi, N., Bansal, R.C., Kumar, J. (eds) Smart Energy and Advancement in Power Technologies. Lecture Notes in Electrical Engineering, vol 926. Springer, Singapore. https://doi.org/10.1007/978-981-19-4971-5_62
13. Breiman, L. 2001. Random Forests. Machine Learning, Vol. 45 Issue 1, pp. 5-32.
14. Bernard, S., Heutte, L., Adam, S. 2009. On the selection of decision trees in random forests. International Joint Conference on Neural Network , pp. 302–307
15. Borgohain, O., Dasgupta, M., Kumar, P., Talukdar, G. (2021). Performance Analysis of Nearest Neighbor, K-Nearest Neighbor and Weighted K-Nearest Neighbor for the Classification of Alzheimer Disease. In: Borah, S., Pradhan, R., Dey, N., Gupta, P. (eds) Soft Computing Techniques and Applications. Advances in Intelligent Systems and Computing, vol 1248. Springer, Singapore. https://doi.org/10.1007/978-981-15-7394-1_28
16. Haykin, S. (2001) Redes Neurais—Princípios e Práticas. 2nd Edition, Bookman, Porto Alegre.
17. Mitchell, T.M. (1997) Machine Learning. McGraw-Hill, New York

Figures

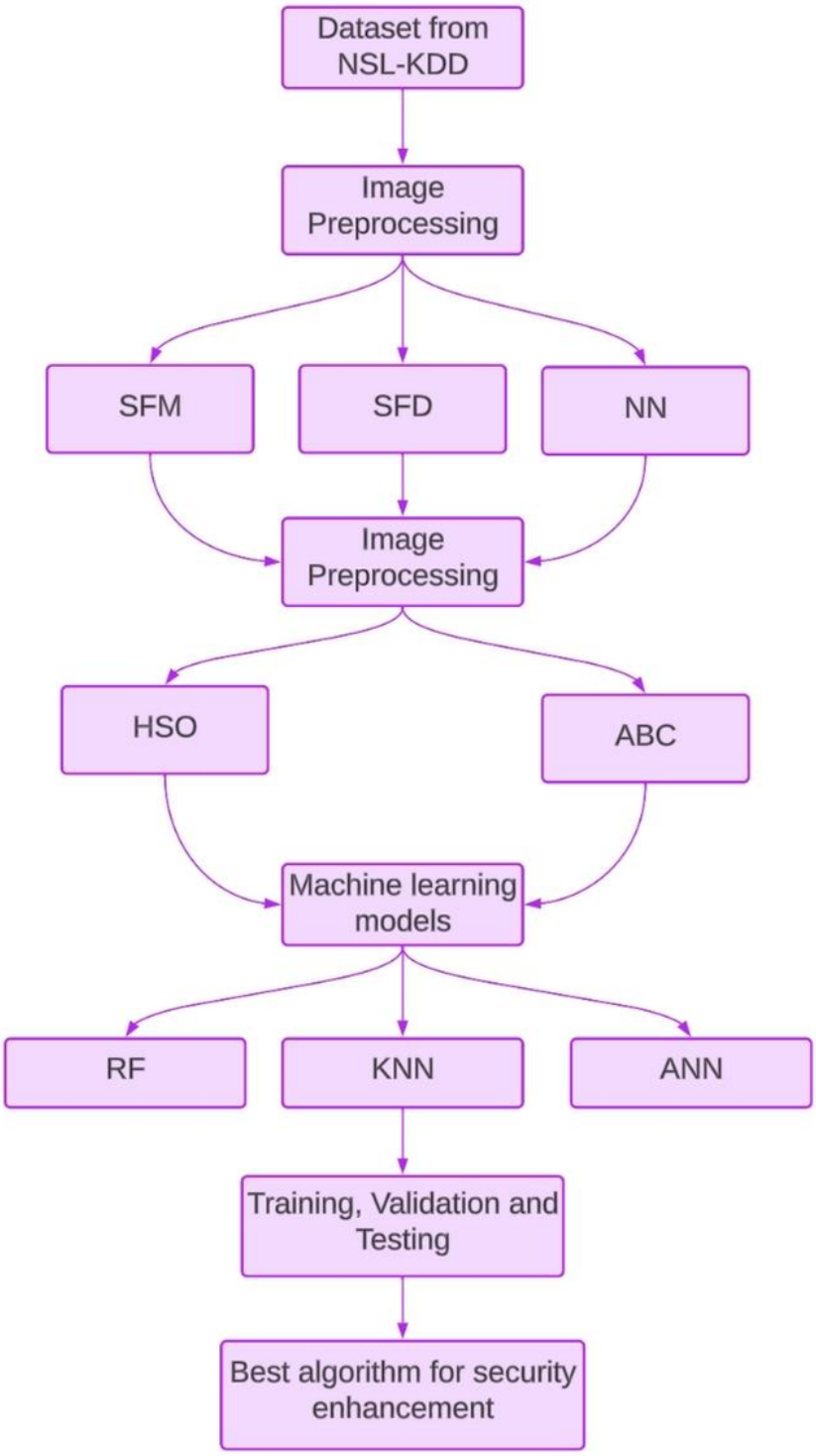

Figure 1

Steps involved in the study

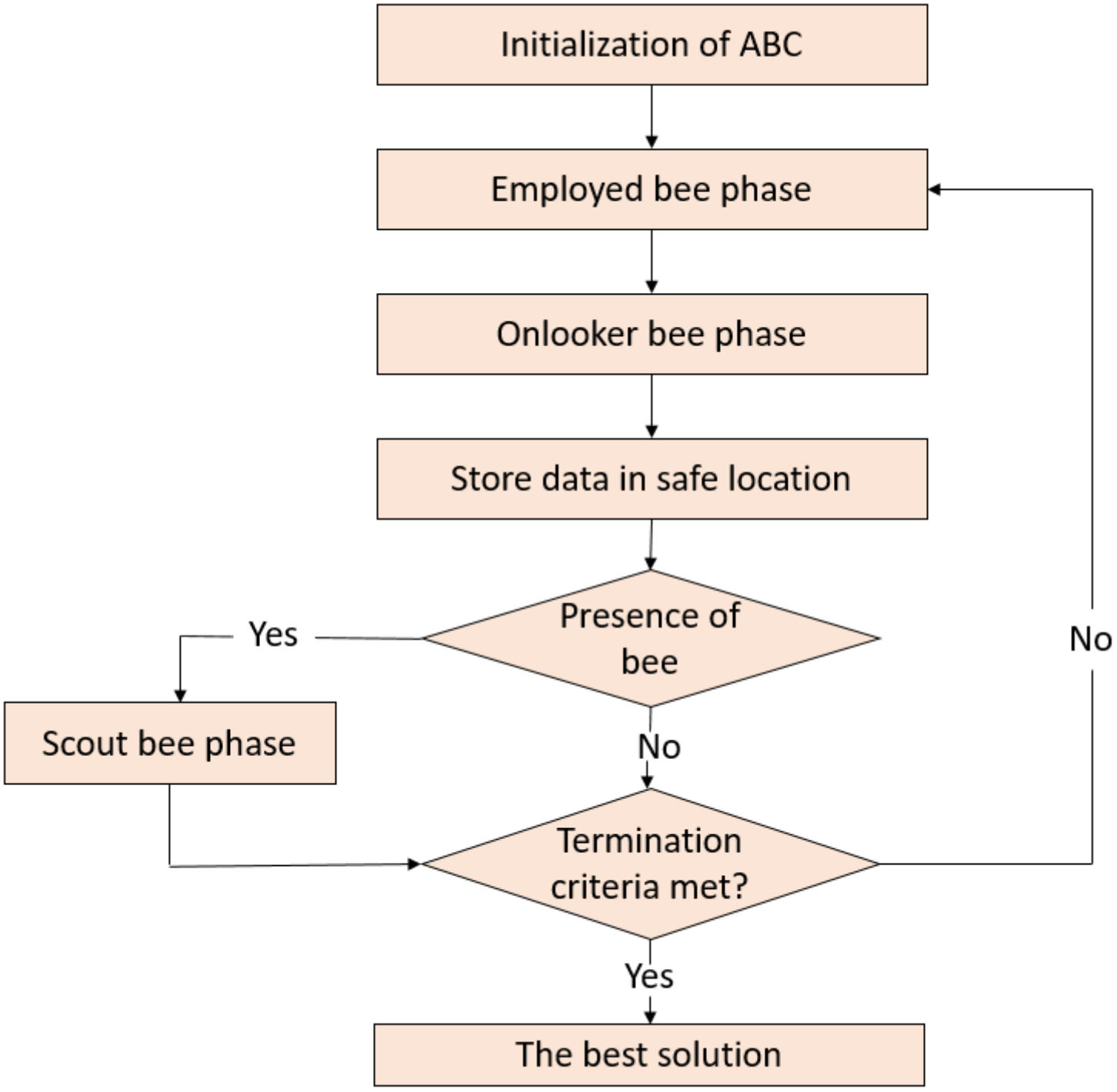

Figure 2

Workflow of ABC

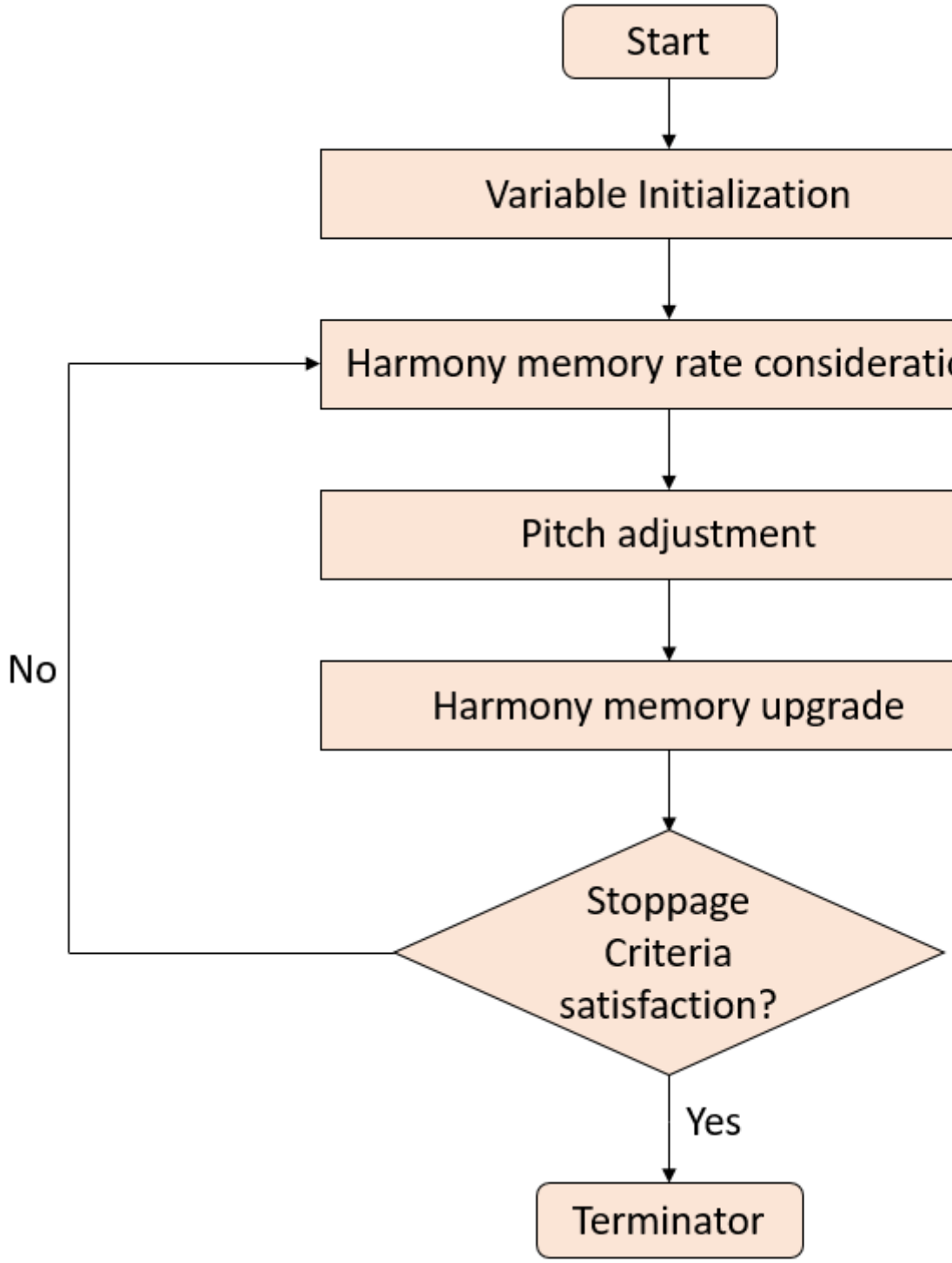

Figure 3

Flow of HSO

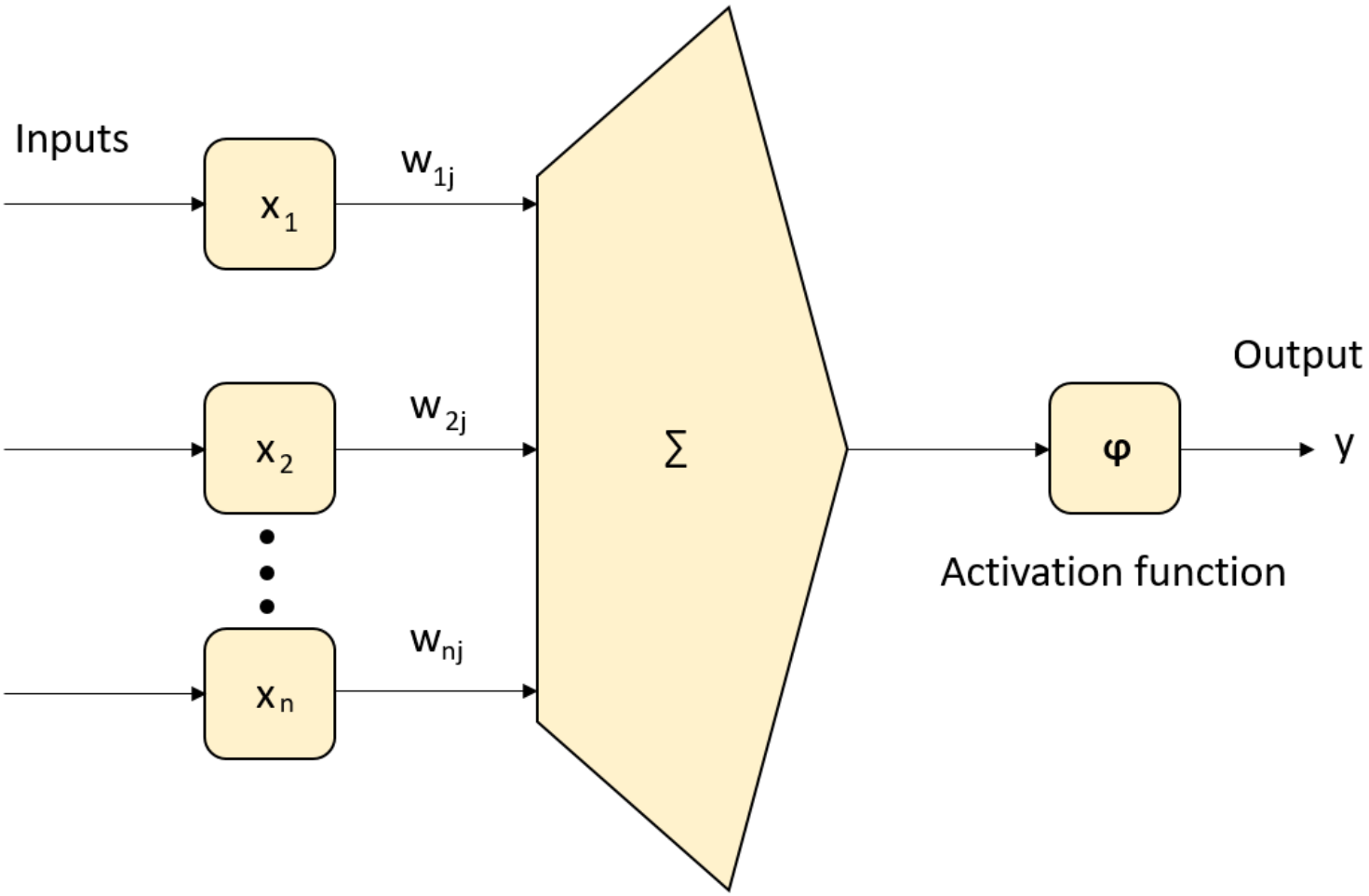

Figure 4

Fundamental components of an artificial neuron

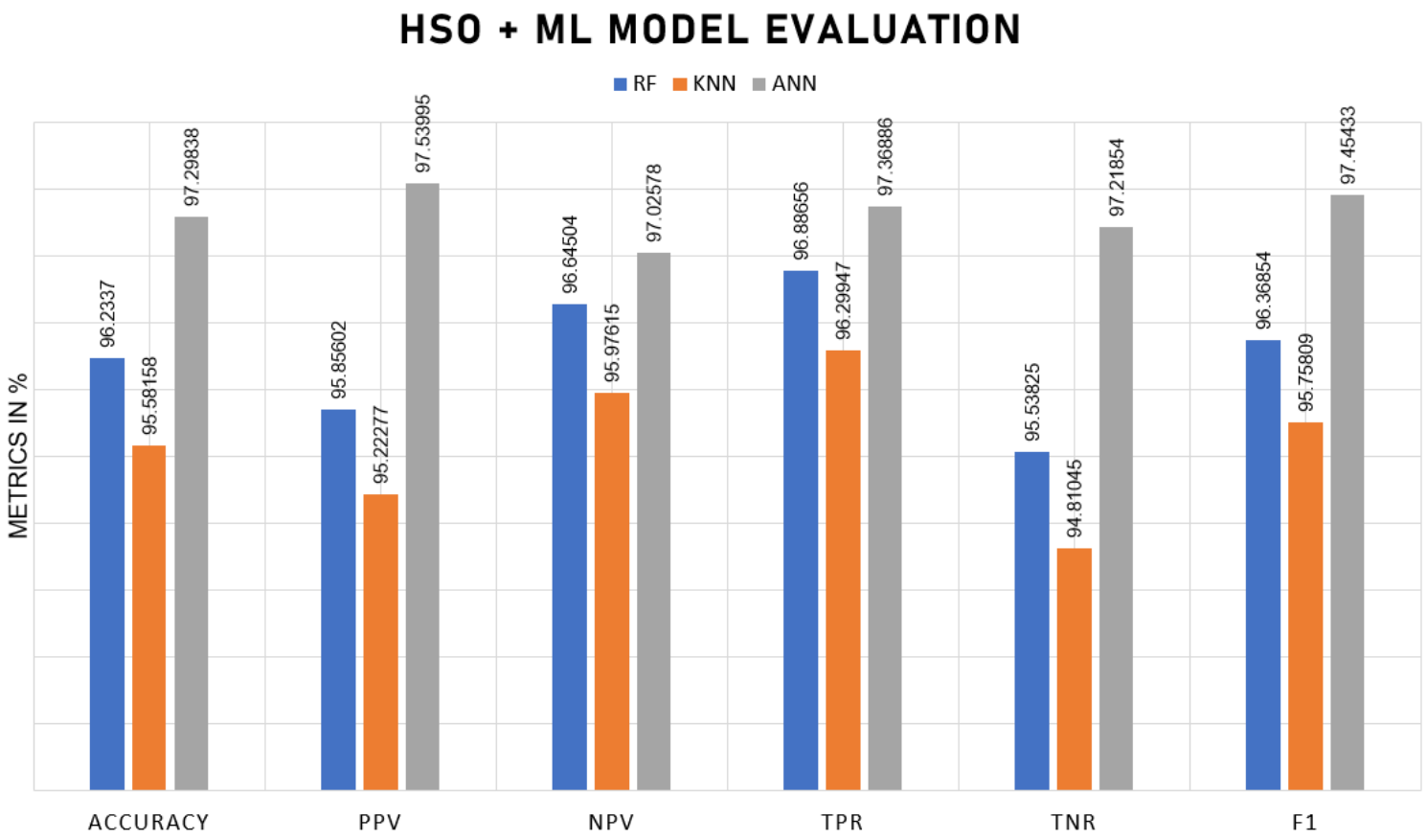

Figure 5

HSO and ML models

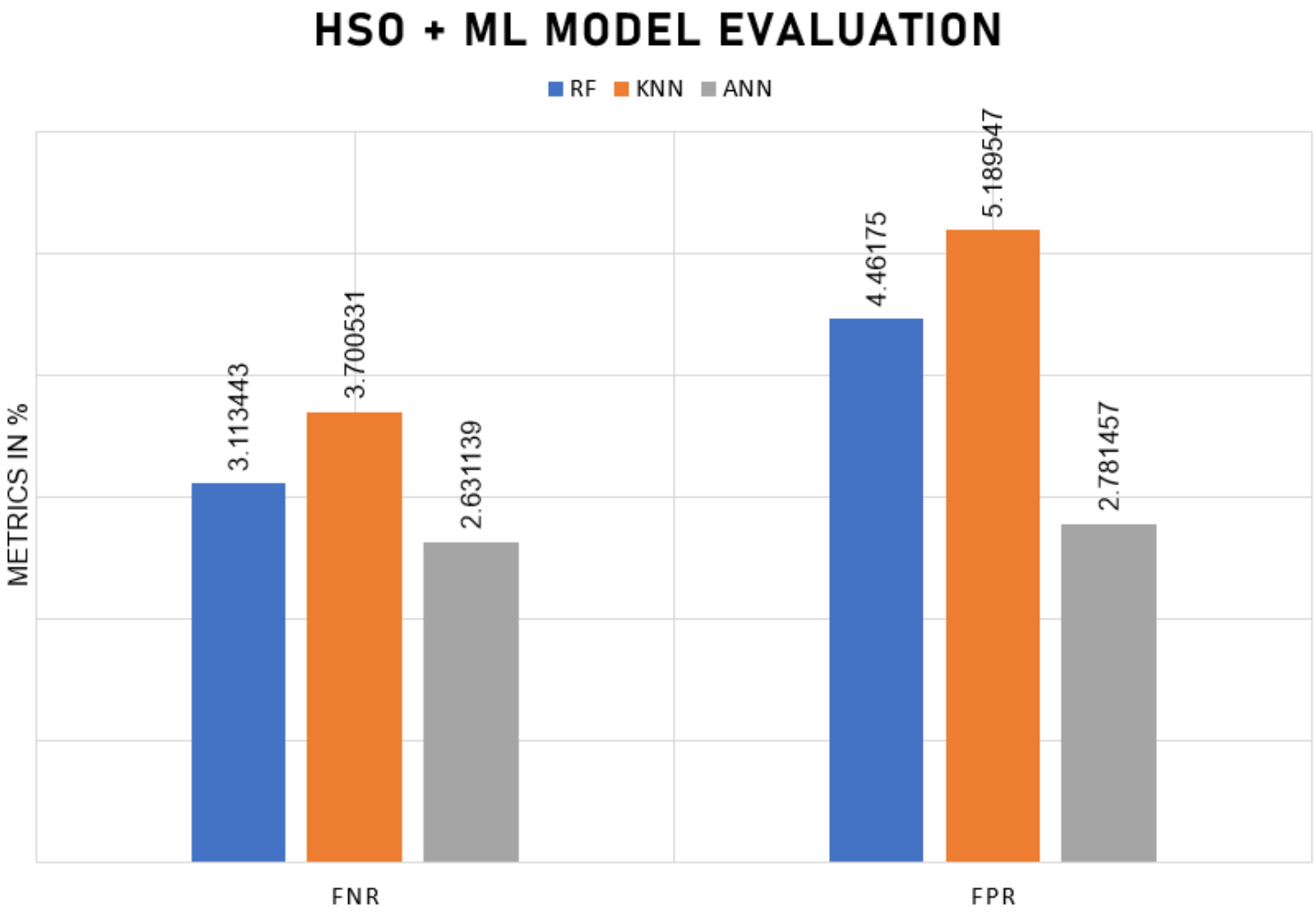

Figure 6

False rates of HSO

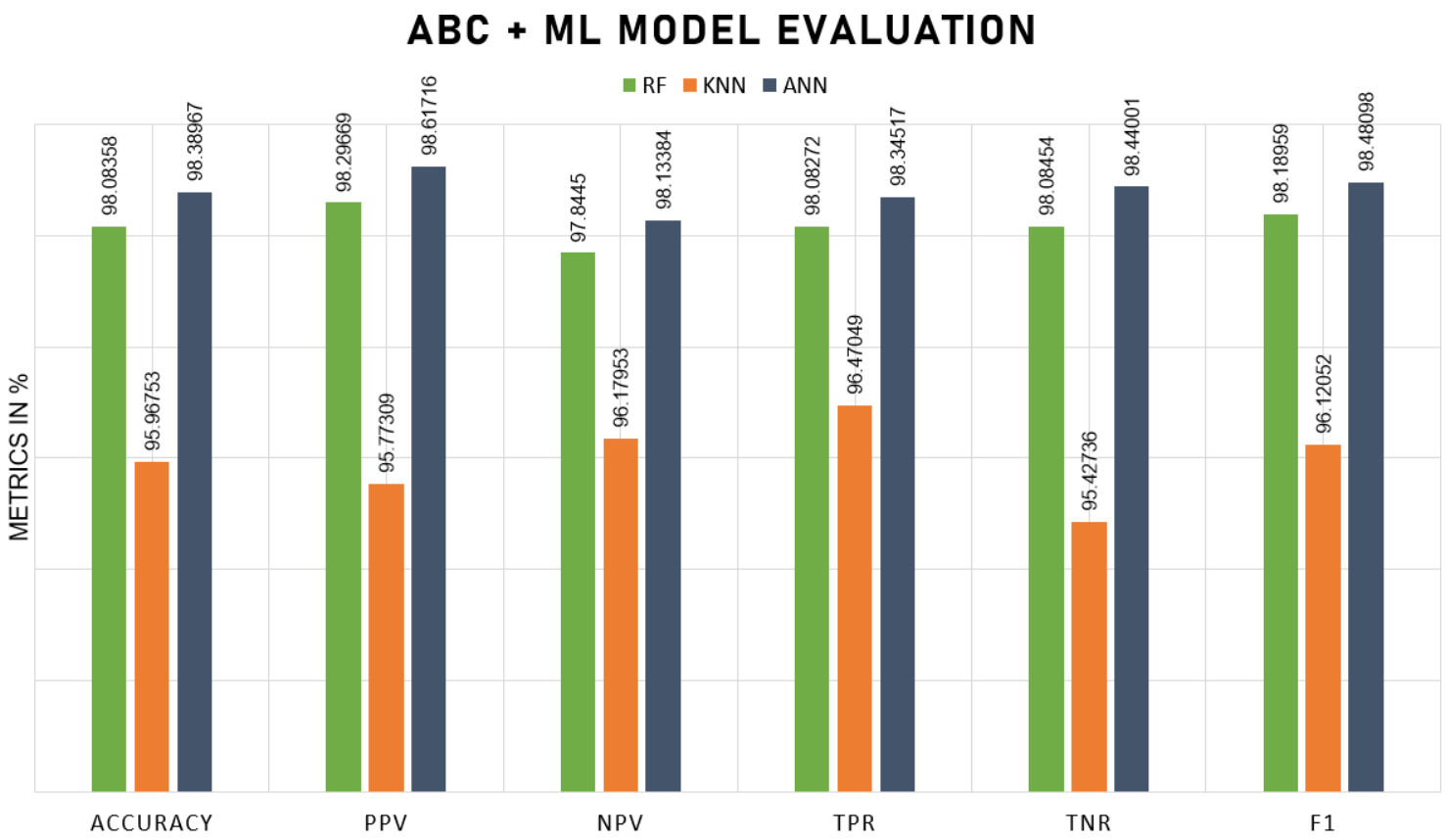

Figure 7

ABC and ML models

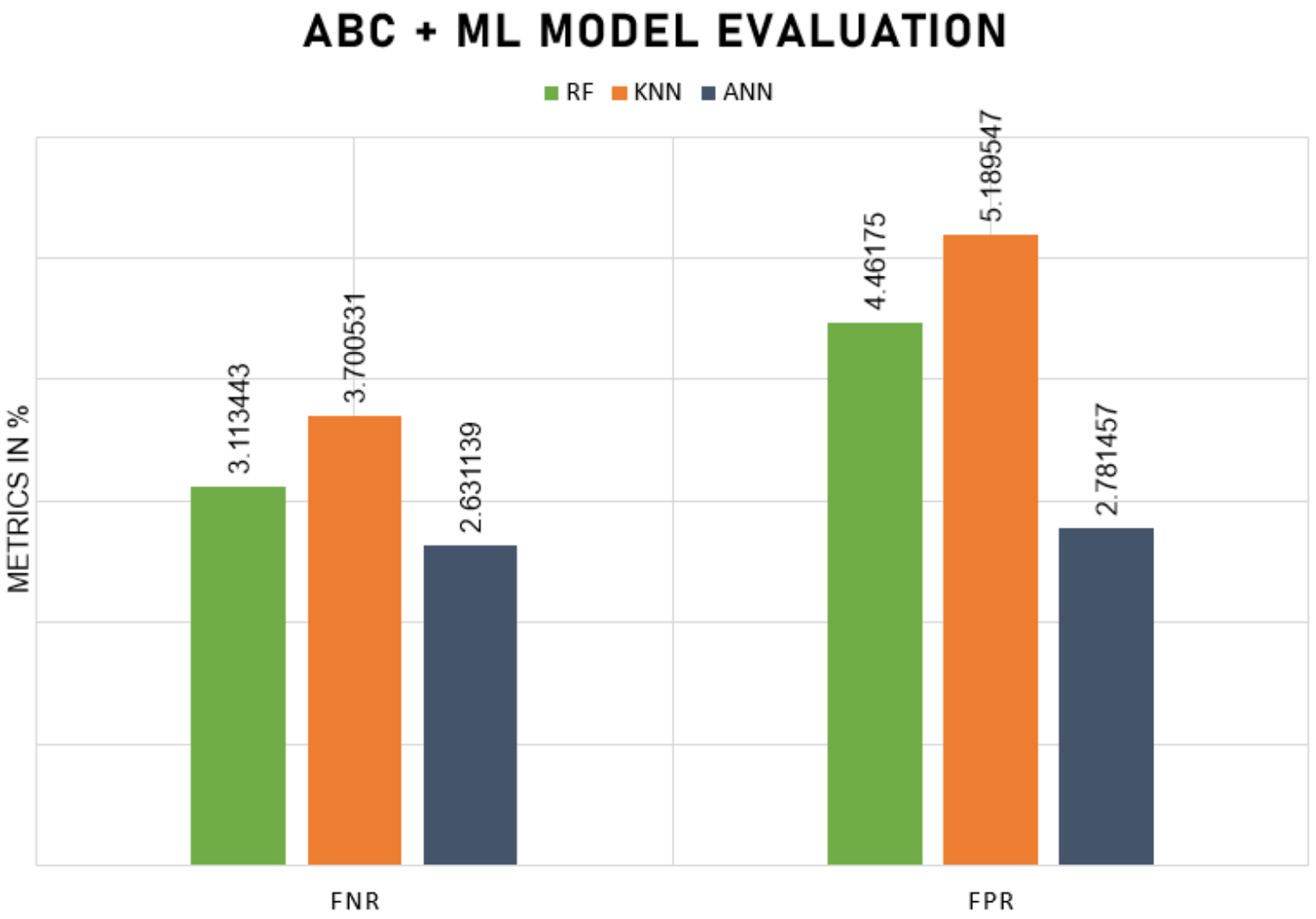

Figure 8

False rates of ABC