Engineering Applications of Artificial Intelligence 114 (2022) 105123

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

ML-based reconfigurable symbol decoder: An alternative for next-generation communication systems

Saurabh Srivastava [*], Prajna Parimita Dash

Department of Electronics and Communication Engineering, Birla Institute of Technology, Mesra, Jharkhand, India

ARTICLE INFO

Keywords:
Machine Learning based reconfigurable decoder
Naïve Bayes
Ensemble Bagged Decision Tree
Decoding performance
PD-NOMA

ABSTRACT

Modern Machine Learning (ML) techniques offer numerous opportunities to enable intelligent communication designs while addressing a wide range of problems in communication systems. A wide majority of communication systems ubiquitously employ the Maximum Likelihood (MLH) decoder in the symbol decoding process with QPSK modulation, thereby providing a non-reconfigurable solution. This work addresses the application of an ML-based reconfigurable solution for such systems. The proposed decoder can be considered a strong candidate for future communication systems, owing to its upgradable functionality, lower complexity, faster response, and reconfigurability. First, a novel low-complexity dataset for model training/testing is generated, that uses only the received symbols. Subsequently, three predictors are extracted from each of the received noisy symbols for model training/testing. The model is then trained/tested using nineteen standard ML-based classifiers, and the computations of various performance metrics indicate the suitability of Naïve Bayes (NB), and Ensemble Bagged Decision Tree (EBDT) classifiers for the model. The simulation results show that the model respectively delivers significant decoding accuracies and error rates of about 93% and 7% during testing, even for a low SNR of 5 dB. Moreover, the statistical analysis of simulation results shows the marginal superiority of the Gaussian Naïve Bayes (GNB) classifier. Further, the model reconfiguration is validated using a BPSK modulated dataset. Finally, a user-separation scheme that eliminates Successive Interference Cancellation (SIC) in the next-generation Power-Domain (PD) Non-Orthogonal Multiple Access (NOMA) networks is suggested by employing the proposed decoder.

1. Introduction

Recently, Artificial Intelligence (AI)/ML techniques are being utilized for various detection-based real-world problems. Applications such as Quadrature Amplitude Modulation (QAM), Spatial Modulation (SM), Generalized Spatial Modulation (GSM), modulation classification, SNR estimation, pulse shaping, Convolutional Neural Network (CNN), and applicability in Long Term Evolution (LTE), Wireless Local Area Network (WLAN), and the next-generation Visible Light Communication (VLC) networks represent their relevancy. AI/ML techniques are significantly helpful in determining the unknown patterns and their influence on the optimizable objective function, so such applications are highly researched in various real-life scenarios. The application of AI in the existing communication systems is still an open area for research, specifically for decoding the received symbols among various modulation schemes. Some ML algorithms are employed for clustering analysis and classification purposes in such AI applications but have not been applied for symbol decoding in the existing communication systems.

QPSK is regarded as a prominent modulation scheme of modern communication systems, that is also being utilized for next-generation communication systems (Tan et al., 2022; Moon et al., 2022), owing to its good trade-off between higher bandwidth utilization and lower error rate. The Fifth-Generation (5G) standards organization, European Telecommunication Standards Institute (ETSI), also mentions the application of the QPSK modulation mapper in the technical specifications of 5G-New-Radio (NR) (Tg, 2018). The QPSK modulation mapper maps a pair of bits $\{b(2i), b(2i+1)\}$ to a modulation symbol in the complex 2-dimensional signal space, $d(i)=\frac{1}{\sqrt{2}}[(1-2b(2i))+j(1-2b(2i+1))]$. At the receiver, this modulation symbol $d(i)$ suffers amplitude and phase variations due to channel noise and gets corrupted as $d\hat{(i)}$. For the current work, the problem of successful decoding is considered, by employing an ML-based alternative to the classical MLH decoder. The present work assumes noise to be zero-mean Additive White Gaussian Noise (AWGN) $\sim\mathcal{N}(0,\sigma^2)$. Now the task of the receiver is to obtain the correct symbol $d(i)$ from the noisy symbol $d\hat{(i)}$.

This can be seen in Fig. 1(a), where the red-colored box shows one of the transmitted symbols $d(i)$ and the nearby green point represents the noisy symbol $d\hat{(i)}$, for a given SNR. The receiver then tries for the best approximation to the transmitted symbol $d(i)$ utilizing the statistical properties of the corrupting AWGN. On increasing the SNR,

* Corresponding author.
E-mail addresses: saurabhnitkian@gmail.com (S. Srivastava), ppdash@bitmesra.ac.in (P.P. Dash).

https://doi.org/10.1016/j.engappai.2022.105123
Received 9 February 2022; Received in revised form 21 June 2022; Accepted 22 June 2022
Available online 1 July 2022

Glossary

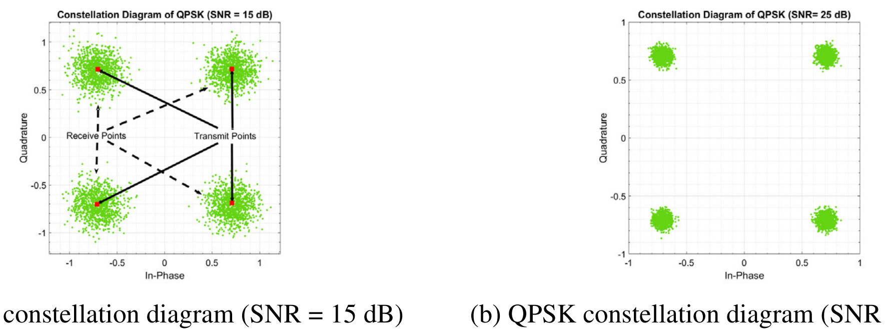
$\mathbf{H}$	Channel Matrix.
FN_n	False negative for class n.
FP_n	False positive for class n.
N	Noise vector (Complex).
P_1	Predictor (complex) obtained from noisy signal.
P_2	Predictor obtained from real-part of P_1.
P_3	Predictor obtained from imaginary-part of P_1.
S	Closed set of QPSK constellation symbols.
TN_n	True negative for class n.
TP_n	True positive for class n.
T	True class of transmitted symbol.
X	Vector of transmitted symbols.
Y	Vector of received symbols.
$d\hat{(i)}$	Received corrupted QPSK symbol.
$\mathcal{N}(0, \sigma^2)$	Gaussian distribution with 0 mean and σ^2 variance.
$d\tilde{(i)}$	Estimated QPSK symbol.
$\{0, 1, 2, 3\}$	QPSK constellation points.
$\{b(2i), b(2i+1)\}$	Input QPSK bits.
$d(i)$	QPSK modulated symbol.
h	Channel Coefficient.

Acronyms

5G	Fifth-Generation.
AI	Artificial Intelligence.
AUC	Area Under the Curve.
AWGN	Additive White Gaussian Noise.
B5G	Beyond 5G.
Bagging	Bootstrap Aggregating.
BPSK	Binary Phase-Shift Keying.
CM	Confusion Matrix.
CNN	Convolutional Neural Network.
DL	Deep Learning.
DNN	Deep Neural Network.
DT	Decision Tree.
EBDT	Ensemble Bagged Decision Tree.
ETSI	European Telecommunication Standards Institute.
FN	False Negative.
FNR	False Negative Rate.
FP	False Positive.
FPR	False Positive Rate.
GNB	Gaussian Naïve Bayes.
GSM	Generalized Spatial Modulation.
IID	Independent and Identically Distributed.
IoT	Internet of Things.
IQR	Inter Quartile Range.
KNB	Kernel Naïve Bayes.
LR	Logistic Regression.
LTE	Long Term Evolution.
MAP	Maximum-A-Posteriori.
MCC	Matthews Correlation Coefficient.
MD	Minimum Distance.
MIMO	Multiple Input Multiple Output.
ML	Machine Learning.
MLbD	ML-based Decoder.
MLH	Maximum Likelihood.
MVMN	Multi-Variate Multi-Nomial.
NB	Naïve Bayes.
NN	Neural Network.
NNR	Nearest Neighbor.
NOMA	Non-Orthogonal Multiple Access.
NR	New-Radio.
PD	Power-Domain.
PPV	Positive Predictive Value.
PSK	Phase-Shift Keying.
QAM	Quadrature Amplitude Modulation.
QoS	Quality of Service.
QPSK	Quadrature Phase-Shift Keying.
RL	Reinforcement Learning.
ROC	Receiver Operating Characteristics.
SC	Superposition Coding.
SDR	Software-Defined Radio.
SER	Symbol Error Rate.
SIC	Successive Interference Cancellation.
SISO	Single Input Single Output.
SM	Spatial Modulation.
SNR	Signal to Noise Power Ratio.
SVM	Support Vector Machine.
SWIPT	Simultaneous Wireless Information and Power Transfer.
TN	True Negative.
TNR	True Negative Rate.
TP	True Positive.
TPR	True Positive Rate.
VLC	Visible Light Communication.
WLAN	Wireless Local Area Network.

the discrete received noisy symbols tend to concentrate on the respective transmitted symbol, as shown in Fig. 1(b).

To decode the correct symbol or the best approximation, various algorithms such as the MLH, Maximum-A-Posteriori (MAP), Minimum Distance (MD), Nearest Neighbor (NNR), and others, exist in the literature. Among them, the MLH is popularly used by researchers (Men and Jin, 2014). For a given received symbol $d\hat{(i)}$, the MLH decoder estimates $d\tilde{(i)} \in \{0, 1, 2, 3\}$, where $\{0, 1, 2, 3\}$ represents one of the QPSK constellation points. In other words, the MLH decoder maximizes the likelihood function or the probability given by $\mathbb{P}\{d\hat{(i)}\ \textit{received} \mid d(i)\ \textit{sent}\}$. As the QPSK modulator has four different constellation points in the complex signal space that represents the transmitted symbol $d_m(i), m = 0, 1, 2, 3$, each of the points can be regarded as being of a specific class. The model proposed in this work assumes the class labels as elements of the set $\{0, 1, 2, 3\}$, corresponding to each of the symbols $d\hat{(i)}$. The MLH decodes, and then determines $d\tilde{(i)} = \underset{0 \le m \le 3}{\arg\min} \| d\hat{(i)} - d_m(i) \|^2$. The NNR decoding of a received symbol $d\hat{(i)}$ also operates by minimizing the Euclidean distance metric between $d\hat{(i)}$ and all the probably transmitted symbols $d\tilde{(i)}$ and gives a similar result. For the QPSK-symbol decoding problem, it becomes intuitive to apply simple ML algorithms and develop a reconfigurable system for the same. As per the author's knowledge, an ML-based reconfigurable model has not been proposed in the literature. This motivates us to apply classical ML algorithms to the existing symbol decoding problem. This paper utilizes the existing classical ML algorithms for the decoding of QPSK symbols at the receiver, considering an AWGN channel.



(a) QPSK constellation diagram (SNR = 15 dB) (b) QPSK constellation diagram (SNR = 25 dB)

Fig. 1. QPSK constellation points with two different SNRs.

The randomness of the received symbols motivates the application of learning algorithms in this work. Specifically, a supervised-learning-based reconfigurable model is developed and validated in this work. Supervised learning maps the input data to the output data, and is extensively used in data classification problems.

Reconfigurability is a growing trend in modern electronics (Lyke et al., 2015), where it provides flexible control through different bit-pattern specifications. A reconfigurable learning-based system shows higher reliability, ease of upgradation, and reduced costs, apart from an embedded intelligent ML algorithm, that motivates its candidature in the next-generation systems. Such systems are highly solicited in the Software-Defined Radio (SDR) platforms. In this paper, a flexible reconfigurable symbol decoder is proposed, and its performance is compared with the existing non-reconfigurable decoder. Specifically, the decoding performances of the EBDT (Ghosh et al., 2021), and NB (Blanquero et al., 2021) classifiers are compared against the MLH decoding performance, for a base system such as QPSK.

Classifiers such as NB (Blanquero et al., 2021) and techniques such as Bootstrap Aggregating (Bagging) (Kotsiantis, 2014; Hillebrand et al., 2021) are used for the same. The specific dataset generated is utilized to make the system learn; hence is referred to as a QPSK ML-based Decoder (MLbD) dataset. Further, to validate the reconfigurability of the model, the same model is evaluated for a BPSK modulation system. Hence, this work presents an interesting and novel application of ML in next-generation communication systems. The contributions of this work are:

- Application of the classical ML algorithms to the QPSK-symbol decoding problem in the existing communication systems
- Development of a reconfigurable symbol decoder that is independent of the underlying modulation scheme; hence can be used for next-generation communication systems
- Generation of smaller QPSK-MLbD datasets for training, validation, and testing of the proposed model, that does not require pre-processing (as in Deep Learning (DL)). All the employed predictors are derived using a single parameter only, i.e., the received noisy symbols
- Performance evaluation of the NB and EBDT decoders through decoding accuracy and symbol error along with the comparative analysis of the MLH decoder
- Model reconfigurability validation through a BPSK modulation scheme, using the generated dataset, specifically for the BPSK modulation

This work considers the novel application of ML algorithms to train the model through the supervised multi-class classification. The trained model is then compared with the existing MLH decoder for its performance. The results show the comparable performance of both the decoding schemes, however, the proposed model is reconfigurable since it utilizes the ML algorithms. Another advantage of the proposed model is its lower complexity and faster operation due to the following reasons. Firstly, the model utilizes only three predictors for symbol decoding. Secondly, the received noisy symbol itself acts as the first predictor, and the other two required predictors are extracted using the first predictor, as explained later in Section 3.1. Thirdly, a lower number of predictors are used in the dataset preparation, which implies much low complexity and faster decoding than any other DL model. Last but not the least, the next-generation communications systems are required to include intelligence in them, and as such the proposed model may serve as a prototype for a low complexity, reconfigurable, fast and intelligent symbol decoder.

The novelty of the proposed work lies in the modeling of the decoding problem through a reconfigurable system. Since, it is crucial to study the system performance under low SNR, as the systems are more error-prone at lower SNRs. The proposed system can perform well, even under low SNR scenarios, and can be utilized for decoding the users' data in next-generation PD-NOMA systems, that currently plan to use the SIC decoding process. SIC and SC are the two processes for such systems, with the former at the receiver side, and the latter at the transmitter side, respectively. The SC and SIC processes are highlighted in Figs. 2(a), and 2(b), respectively. As per the authors' knowledge, the PD-NOMA networks employ SIC to differentiate between the users' messages, and as such the limitations of SIC viz. error-propagation and higher latency pose stringent system restrictions; however, the proposed model overcomes these limitations by eliminating the SIC.

The rest of the paper is organized as follows. A literature survey of some prominent ML-based techniques, and their application in modern as well as state-of-the-art communication systems, is given in Section 2. This survey motivates an ML-based model that finds application in the next-generation NOMA networks. Section 3 describes the proposed model and provides details on the methodology adopted for proper dataset generation, model training, validation, and finally its testing using simulations in MATLAB. A discussion on the performance measures related to this work followed by a discussion on the simulation results obtained is presented in Section 4. Finally, the paper concludes by highlighting the future scope of the work in Section 5.

2. Related work

ML algorithms utilize statistical, probabilistic, and optimization approaches to learn from previous experiences and discover valuable patterns in vast, complicated datasets. Primarily, these are used for classification and regression purposes. Classification implies the categorization of instances, which may be in two classes (binary), or more than two classes (multi-class). The NB and the Decision Tree (DT) are binary classifiers that may be utilized for multi-class classification. The binary and multi-class applications of these algorithms range from automated text categorization (Bhavani and Kumar, 2021; Kukreja et al., 2021; Charalampakis et al., 2016), information technology (Meng et al., 2022), consumer purchase behavior recognition (Wang and Xu, 2020), industrial processes (Li et al., 2020; Shen et al., 2021), face detection (Soula et al., 2020), pollution modeling (Ostad-Ali-Askari et al., 2017), disease modeling (Park et al., 2021; Ghosh et al., 2021), etc. An

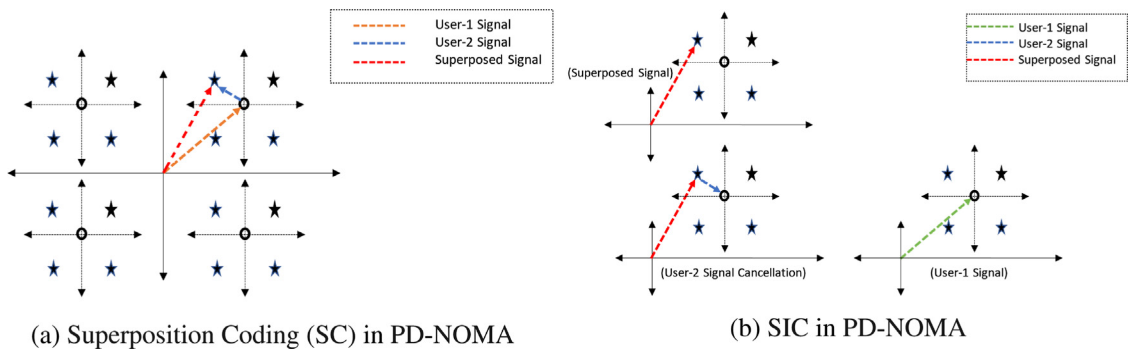


Fig. 2. Illustration of SC and SIC processes.

interesting application domain of such algorithms is the communication area. Oswaldo Simeone discussed the application of some ML-based algorithms in communication systems (Simeone, 2018). Further, the scope, limitations, and future enhancements of ML techniques are analyzed from a 5G/Beyond 5G (B5G) viewpoint (Morocho-Cayamcela et al., 2019). The authors define the supervised classification problem formulation using instances in the training dataset and the corresponding labels. Then the ML algorithm finds a functional mapping from the training dataset instances to the corresponding labels. Moreover, the authors define the classification performance measure of accuracy as the percentage of samples for which the model produces the correct output.

Moreover, the supervised ML approach for specifically shaped pulse signals has been studied (Bari et al., 2016). The authors analyzed the accuracy performance of various classifiers and showed a decent performance of an ML classifier for their application. A comparison of SNR estimation techniques for BPSK in the real AWGN channel and 8-Phase-Shift Keying (PSK) in the complex AWGN is also performed (Pauluzzi and Beaulieu, 2000), mentioning the MLH estimator as the optimum estimator for the given application. Further, a low complexity MLH detection algorithm with an M-ary PSK constellation has been suggested (Men and Jin, 2014). The proposed detector's Symbol Error Rate (SER) performance is compared with the optimal MLH-SER performance, with lower computational complexity. Some other state-of-the-art applications of a few ML-based algorithms are provided (Xie et al., 2020; Chen et al., 2021). On the other hand, a DT is used for classification using the discrete target class values, being a popular ML algorithm owing to its simplicity (Wu et al., 2008). This paper uses the ensemble method of bagging, for achieving a higher decoding accuracy and a uniform prediction. DTs have also been used for medical and academic performance predictions (Njoku, 2019). Thus, an ensemble of DTs is employed for a uniform decision, by constructing subsets of the training data, evaluating their decision, and finally voting on the trees (Breiman, 1996).

Similarly, the optimality of the NB classifier is explored (Zhang, 2005; Kupervasser, 2014). Kotsiantis et al. (2007) mentions prediction accuracy as the performance metric for classifier evaluation and reports cross-validation techniques for effective evaluation of the classifier performance. In this paper, five-fold cross-validation of the training dataset is performed, by dividing the training set into five mutually exclusive, equal-sized subsets. For each subset, the classifier is trained on the union of all other subsets, and the error rate of each subset is computed, representing the error rate of the classifier. The application of such ML algorithms usually demands high prediction accuracy, as well as low misclassifications cost.

A few recent applications of ML are included in the Internet of Things (IoT) networks, for improving the Quality of Service (QoS) (Vairagade and SH, 2021), in the Simultaneous Wireless Information and Power Transfer (SWIPT) NOMA network (Sparjan et al., 2021). Similarly, Bayesian Neural Network (NN) has been employed for attack detection on IoT devices (Toğaçar, 2022). Further, a vertical handover decision is evaluated using fuzzy logic for the decision process (Drissi et al., 2017). The next-generation massive-connectivity issues and spectrum scarcity highlight the application of NOMA, even under generalized fading conditions (Nauryzbayev et al., 2021).

Different from the work (Pauluzzi and Beaulieu, 2000), the model proposed in this paper does not compute the SNR, but instead decodes the received noisy symbols, independent of the underlying modulation scheme; hence, it is flexible and therefore suits the next-generation communications. Using the AI-based approach, the complexity becomes independent of the number of antennas, against the optimal MLH approach adopted (Men and Jin, 2014). Similarly, the work (Xie et al., 2020) utilizes DL-based techniques that require heavy resources from the end-user. The authors (Xie et al., 2020) derived a DL-based SNR estimation technique with three DL networks — AlexNet, Inception V1, and VGG 16, respectively, and compared their performances. Specifically, the SNR is estimated utilizing the three-channel constellation diagrams by mapping signal observations to a 7×7 complex plane and using a significant number of predictors for the prediction. This increased the training times to many hours, which is quite undesirable for a reconfigurable system. Further, as our model requires only the received symbols and not a massive number of images, our approach can be substituted for DL-based SNR detection, as a future job. Inspired by the works (Morocho-Cayamcela et al., 2019), our model helps to reduce the error probability by replacing the SIC block in the next-generation NOMA network, using an SDR setup (Garnier et al., 2020). Another application is achieving a reduced decoding-error probability through resource allocation in future networks (Qureshi et al., 2021), through a reconfigurable system. However, this will be a future extension of the present work. A few specific contributions towards the application of AI/ML techniques in communications are represented in Table 1. A majority of the considered references highlight the presence of DL, NN, and similar techniques. However, no such reference deals with the symbol decoding problem and its implementation in the next-generation communication systems.

3. Proposed ML-based reconfigurable decoder

The central task of the work is to estimate the transmitted QPSK symbol $d(i)$ from the corrupted received symbol $\hat{d(i)}$ at a given SNR. For this, the formulated problem becomes an estimation problem (estimating the correct QPSK symbol from the interference-plus-noise corrupted symbol). Thus, it is required to determine the mapping function from the received symbol to the transmitted symbol.

The block diagram of the research is shown in Fig. 3, and the model considered in this work is shown in Fig. 4, where a performance comparison is made between the existing MLH decoder and the proposed ML-based decoders, i.e., KNB, GNB, and EBDT decoders, and it is suggested to replace the MLH decoder with a reconfigurable ML-based decoder. This requires a large collection of data for every SNR value; so the first part of the work is the generation of a suitable dataset for the experimentation, as described in Section 3.1. A similar work (Xie et al.,

Table 1
AI applications in communications.

S. No.	Field of application, Ref. No.	Year	Employed technique
1	QAM/PSK symbol synchronization (Matta et al., 2019)	2019	Reinforcement Learning (RL) agent design through Q-learning
2	Modulation classification (Peng et al., 2021)	2021	Signal representation and data preprocessing in DL
3	DL applications in communications (Liao et al., 2020)	2020	DL in intelligent communication systems, channel estimation, signal detection, and modulation recognition
4	Optical performance monitoring and modulation format identification (Saif et al., 2020)	2020	Modulation Format Identification, Optical Performance Monitoring using DL algorithms
5	Modulation classification Using CNN (Peng et al., 2017)	2017	CNN based DL
6	Signal detection in the GSM-VLC system (Sun et al., 2021)	2021	Support Vector Machine (SVM) aided signal detection
7	Channel estimation and signal detection (Liu et al., 2022)	2022	Tiny ML model
8	Blind modulation classification (Norolahi and Azmi, 2021)	2021	Combined ML and Feature Extraction
9	Communication system (Simeone, 2018)	2018	Supervised and Unsupervised ML algorithms
10	5G/B5G communications (Morocho-Cayamcela et al., 2019)	2019	RL, Q-learning, clustering, NN, Deep Neural Network (DNN), etc.
11	Pulse shaping in communications (Bari et al., 2016)	2016	SVM, Logistic Regression (LR), NN
12	SNR estimation techniques (Pauluzzi and Beaulieu, 2000)	2000	MLH detection
13	SM (Men and Jin, 2014)	2014	MLH detection
14	SNR estimation using constellation diagrams (Xie et al., 2020)	2020	DL
15	IoT attack detection (Toğaçar, 2022)	2022	Bayesian NN
16	LTE and WLAN network selection (Drissi et al., 2017)	2017	Multicriteria decision framework

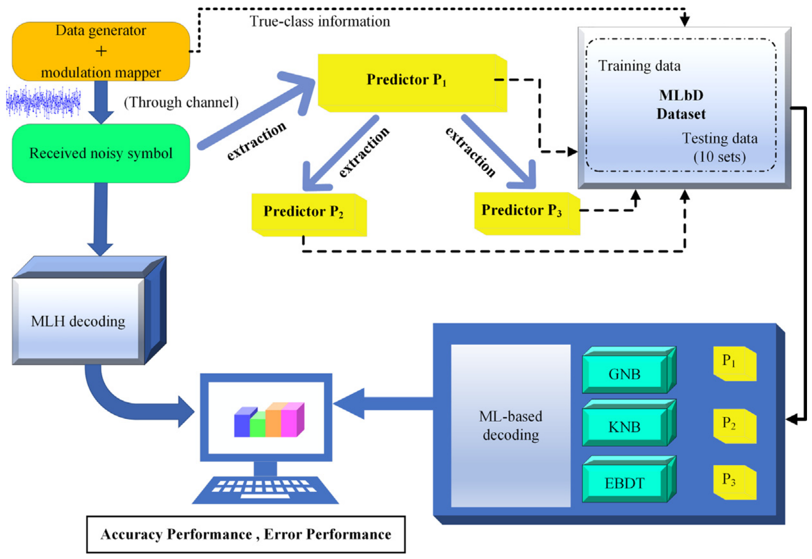


Fig. 3. Block diagram of the proposed research.

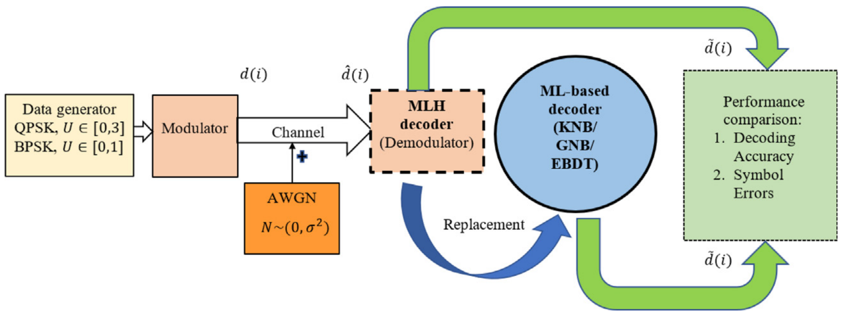


Fig. 4. The proposed ML-based model.

2020), deals with a DL-based SNR estimation technique using three DL networks — AlexNet, Inception V1, and VGG 16, respectively, and compares their performances. Specifically, the estimated SNR is determined by utilizing a three-channel constellation diagram by mapping signal observations to a 7 × 7 complex plane and using a significant number of predictors for the prediction of SNR (Xie et al., 2020). This increased the training times to many hours, which is undesirable for any desired reconfigurable system.

DL techniques, in general, are much more resource-hungry and require huge processing capabilities, and thus are highly complex. To keep the dataset complexity and model training time low, this paper focuses on the generation of simpler QPSK-MLbD datasets for efficient symbol decoding. Another drawback with the DL techniques is the signal pre-processing requirements, such as resizing images to proper resolution for the input layer of the DL network. On the other hand, this work utilizes much smaller QPSK-MLbD datasets, that incorporate very few predictors, against the huge datasets required in DL-based techniques. Further, all the predictors in the generated dataset are extracted using only the received symbol. Since it is assumed that the proposed model utilizes the AWGN channel, each of the QPSK-MLbD datasets is generated for specific values of the channel SNR. In this paper, though all the datasets are generated using MATLAB simulations, the real-field data would also serve the purpose. A complete block diagram of the carried work is shown in Fig. 3, which shows the dataset generation, as well as the application of the dataset to the proposed reconfigurable model.

For this work, a Single Input Single Output (SISO) communication model is considered, which is given as:

$$Y = hX + N \tag{1}$$

where, Y, h, and X, represent the output vector of the received symbols, the channel coefficient between the transmitter and the receiver, and the input vector of the transmitted symbols, respectively. As mentioned earlier, the complex noise vector N is assumed to be Gaussian $\sim \mathcal{N}(0, \sigma^2)$. For the simplest case (without fading or channel effects), the input–output relation is given by:

$$Y = X + N \tag{2}$$

Modified datasets in the presence of channel effects for the Multiple Input Multiple Output (MIMO) are also generated easily by replacing h with a suitable channel matrix $\mathbf{H}$ in Eq. (1), but this paper focuses on highlighting the decoding methodology; so the SISO model specified by Eq. (2) is employed in the paper. QPSK is the simplest modulation format that provides higher reliability by providing admirable robustness even in the presence of high interference; so, a QPSK constellation is selected for this work. The points in set $S = \{00, 01, 10, 11\}$ of the QPSK constellation are assumed to be arranged in a counter-clockwise fashion. Thus, $S = (00)$ represents the red-colored box at the top-right corner of Fig. 1(a) (in the first quadrant). Similarly, $S = (01)$, $S = (10)$, and $S = (11)$, are in the second, third, and fourth quadrants of the complex constellation plane, respectively. Each of the received constellation points is also mapped to the "true" class, $T = \{0, 1, 2, 3\}$, depending upon the quadrant in which it lies.

Using the true-class information given above, and extracting the predictors as shown in Fig. 3, the appropriate dataset is generated, as shown in the next section. As per the author's knowledge, such a novel mechanism for the suitable dataset generation with predictors' extraction using the received symbols, and the application of the generated dataset in a reconfigurable, supervised-learning network, has not been explored in contemporary literature. A major advantage of using this supervised-learning approach in current communication systems is the elimination of complex channel estimation techniques; and in the next-generation PD-NOMA systems, it is the simultaneous decoding of user information, reducing the network latency with the elimination of channel estimation and SIC decoding procedures.

The model considered in this work is represented in Fig. 4. As shown in the figure, the theme of this work is to compare the decoding performances of the existing MLH decoder with those of the proposed reconfigurable learning-based decoders using the generated datasets (shown in Fig. 3), hence, suggesting a novel reconfigurable application of ML in communication systems, especially the next-generation NOMA networks. The SIC decoding algorithm is described in Pei et al. (2022) using an MLH decoder, employing the composite constellation diagram of two NOMA users. However, the ML-based decoder that employs a suitable dataset (similar to MLbD, but prepared for the joint constellation) can decode both the users simultaneously without the SIC process. This offers the possible next-generation system large flexibility, in terms of system reconfiguration.

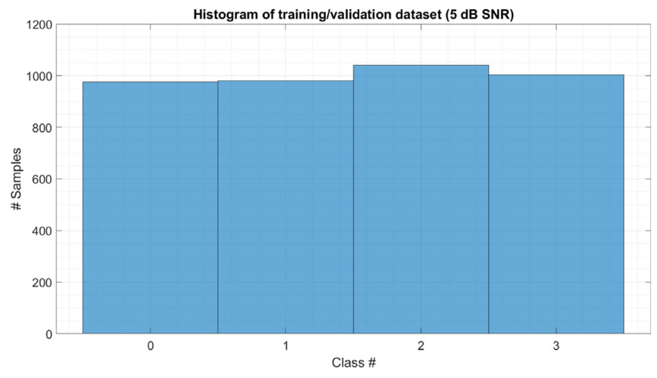


Fig. 5. Typical histogram of generated samples for validation/testing.

3.1. Generation of QPSK-MLbD datasets

The QPSK MLbD datasets used in this work are generated in MATLAB. For the QPSK decoding problem, a dataset of size $64\,000 \times 5$ is generated, out of which 50% of the data is utilized for model-training and validation, and the rest is utilized for a series of ten model tests. The generation of the dataset is shown in Fig. 3. As shown in the figure, the three predictors are extracted using only the received noisy symbol and are used along with the true-class information to form the dataset. Each of the training and validation QPSK-MLbD datasets consists of 4000 complex-valued samples for a specific SNR, distributed evenly among the four true classes. This uniform distribution across the true classes is necessary so that the algorithms do not bias towards a given class. The QPSK-MLbD dataset is generated by corrupting the transmitted symbol $d(i)$ with the AWGN, to produce the received noisy symbol $\hat{d(i)}$, corresponding to the specific SNR.

Initially, 4000 real integer data samples are generated that are uniformly distributed in [0,3], as shown in Fig. 5. As can be seen from the figure, each class includes approximately 1000 samples out of 4000 total validation/training samples. The generated (real) integer data samples are next mapped to the corresponding QPSK constellation points (classes), to yield a complex QPSK symbol $d(i)$. As shown in Fig. 6, each of the 4000 real integer points is mapped to one of the true classes, where the classes are represented by the 2-dimensional complex points $(0.7071 + 0.7071i, -0.7071 + 0.7071i, -0.7071 - 0.7071i, 0.7071 - 0.7071i)$, respectively. The symbol power of the mapped data is computed before the addition of AWGN to ensure a specific transmit SNR. This is next followed by corrupting each of the mapped points with AWGN at the given SNR, to give the noisy QPSK symbol $\hat{d(i)}$, as shown in Fig. 7.

With the MLH decoder, the noisy QPSK symbol is then demodulated, followed by the prediction of its respective class. Comparing the transmitted symbol class and the predicted class of the received symbol, the MLH decoding accuracy can be computed. Other validation datasets for different SNR values are generated following the same procedure. Next, three predictors are employed to form the Multi-Variate Multi-Nomial (MVMN) QPSK-MLbD datasets. These predictors are obtained utilizing only the received noisy QPSK symbol $\hat{d(i)}$. The first predictor P_1 is selected as the complex received symbol $\hat{d(i)}$ itself, and the second predictor P_2 is derived from it, by considering the real part of P_1. Similarly, the imaginary part of $\hat{d(i)}$ is considered the third predictor P_3. This also makes the datasets flexible for use with other 2-dimensional constellation systems.

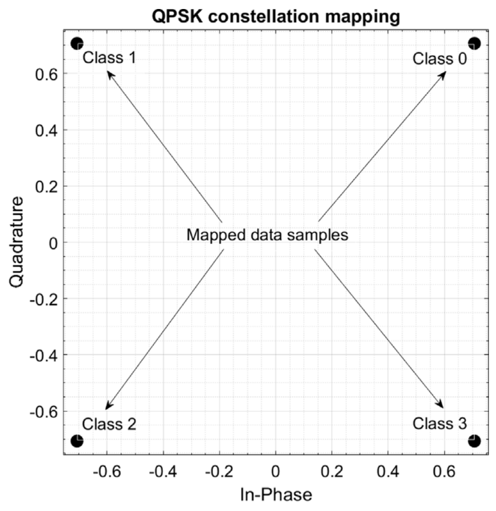


Fig. 6. Mapping of dataset samples into predictor classes of QPSK constellation.

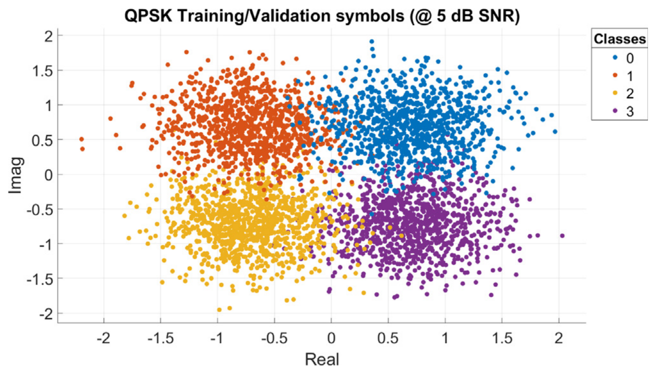


Fig. 7. Generated QPSK dataset symbols for training/validation @ 5 dB SNR.

Table 2
Dataset format.

Predictor P_1	Predictor P_2	Predictor P_3	True class
0.6788 − 1.1342i	0.6788	−1.1342i	3
1.2017 + 0.2474i	1.2017	0.2474i	0
−0.6309 − 0.2956i	−0.6309	−0.2956i	2
…	…	…	…

Table 2 shows the instances of the received symbol and the formulation of predictors with it, as a part of the QPSK-MLbD dataset. Here, the predictor P_1 is the noisy QPSK symbol $\hat{d(i)}$ at the given SNR, and predictors P_2 and P_3 are the real and imaginary parts of P_1, that are used for classifier learning. The last column represents the true class of the symbol $\hat{d(i)}$, which is obtained using Fig. 6. The same format is used to generate the datasets for training and validation, and the test dataset preparation, for specific SNR values. As the generated samples are independent and AWGN is also random, the predictor instances may be considered Independent and Identically Distributed (IID). This independence allows for the use of NB-based classifiers in the model. The NB algorithm leverages the Bayes theorem and assumes that predictors have conditional independence for the given class. Further, a GNB classifier works with continuous predictor values. In this paper, another NB classifier, KNB, which has a Gaussian kernel smoother with unbounded support is employed for the performance evaluation of the proposed ML-based decoder.

For testing the model, 10 test datasets, each of size 3200 × 5, are obtained from the testing part of the QPSK-MLbD dataset. However, keeping a 10:1 ratio of the training and testing dataset samples, each of the test datasets for the given SNR includes 400 samples. Since supervised classification is used in this work, both the training and testing datasets are recorded with the transmitted or true symbol points, to aid in the learning process of the classifiers. It is assumed that the channel statistics do not vary during the period from symbol transmission to symbol prediction at the receiver. For a fair comparison with the extensively used MLH method, the predicted class of the received samples is also determined, along with the symbol detection errors (i.e., misclassifications). The model performance results are tabulated against each SNR value. To effectively evaluate the classifier's performance, five-fold cross-validation is employed, as suggested (Kotsiantis et al., 2007).

Table 3
List of ML-based classifiers for model evaluation.

S. No.	Classifier	S. No.	Classifier
1	GNB	10	Bilayered NN
2	KNB	11	Linear SVM
3	EBDT	12	Coarse Gaussian SVM
4	Wide NN	13	Cubic SVM
5	Coarse Tree	14	Ensemble RUSBoosted Tree
6	Fine Tree	15	Fine Gaussian SVM
7	Medium Tree	16	Medium Gaussian SVM
8	Ensemble Boosted Tree	17	Quadratic SVM
9	Medium NN	18	Trilayered NN
		19	Narrow NN

Table 4
Classifier parameters for simulation.

S.No.	Classifier	Classifier settings
1.	GNB	Numerical predictors distribution: Gaussian; Categorical predictors distribution: MVMN

2.	KNB	Numerical predictors distribution: Kernel;
		Categorical predictors distribution: MVMN;
		Kernel type: Gaussian;
		Support: Unbounded

3.	EBDT	Ensemble method: Bag;
		Learner type: DT;
		Max. no. of splits: 1;
		No. of learners: 211;
		Max. no. of predictors to sample: 2

3.2. Model training and validation

The datasets generated in Section 3.1 are then provided to the classifiers for training and validation purposes. Depending upon the size of the generated dataset and the extracted predictors, which are three in the current work, and the ease of interpretation, a wide variety of ML classification algorithms are selected for model training and validation, ranging from NB to NN to SVM to DT based classifiers. The size of the dataset, and parameters such as training speed, prediction speed, decoding accuracy, and misclassification cost, further motivated the classifier choices for training and validation, in the present work. The training is performed with nineteen classical ML classifiers, out of which only three classifiers (GNB, KNB, and EBDT) showed decent performance for the symbol decoding problem. All the other selected classifiers are not considered owing to their inferior performances in terms of their validation accuracy/cost/prediction speed and/or required training time. The complete list of considered classifiers is shown in Table 3. The classifier parameters considered for simulations with the three outperforming classifiers (with their settings) are given in Table 4.

For the GNB classifier, a Gaussian predictor distribution is assumed, and for the KNB classifier, a Gaussian type kernel distribution is assumed, as can be seen from Table 4. For both the NB classifiers, the categorical predictors' distribution is assumed to be MVMN. For the EBDT classifier, a DT is used for the classification with a bagged ensemble having 211 learners and a single split. The maximum no. of predictors sampled is 2. Table 5 represents the classification cost

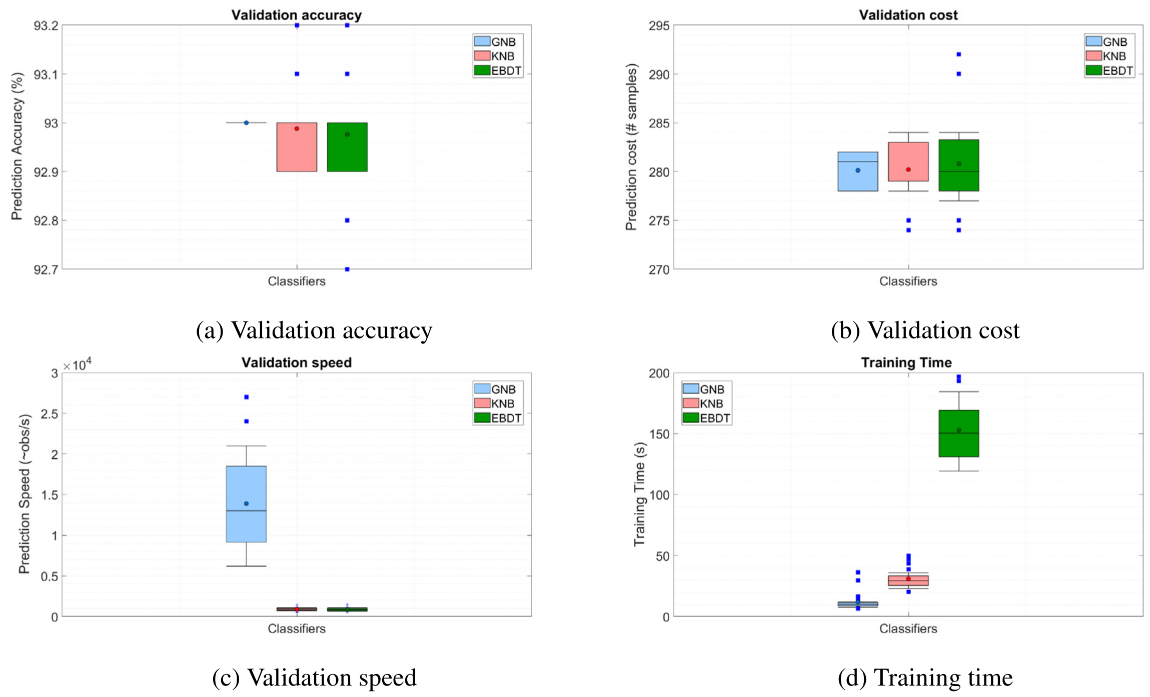


Fig. 8. Training/validation results of classifiers (SNR = 5 dB).

Table 5
Classification cost matrix.

		Predicted class			
		0	1	2	3
True class	0	0	1	1	1
	1	1	0	1	1
	2	1	1	0	1
	3	1	1	1	0

matrix for the system, as every incorrect prediction is given a cost of unity. The system is trained with the prepared dataset of 4000 symbols uniformly distributed over each class (for SNRs 5 dB, 6 dB, …, 12 dB), followed by its validation, using the classifiers mentioned in Table 4. The generated training/validation symbols are shown in Fig. 7. The blue, red-orange, yellow, and violet-colored points, represent classes “0”,“1”,“2”, and “3”, respectively. As shown in the figure, 4000 QPSK symbols are generated with an SNR of 5 dB. The constellation symbols follow a uniform density function, as mentioned earlier.

The training/validation dataset is first split into five disjoint sets, representing a five-fold split for cross-validation. Next, the classifiers are trained with four of the disjoint sets, keeping the fifth one for validation. The process is repeated keeping each disjoint set for validation and using the other four sets for training. Finally, the validation results are averaged over all the disjoint sets to produce a consistent result.

Table 6 shows the performance of GNB, KNB, and EBDT classifiers for 25 runs of training/validation over a given training/validation data. It can be seen that all the three considered classifiers are almost 93% accurate and have around 280 misclassified samples. However, for every validation run, the prediction speed and training time does vary. Table 7 gives the average performance of the three classifiers over the complete training process.

The simulation results indicate good accuracies with the GNB, KNB, and EBDT classifiers, approximately having the same misclassification cost. However, the prediction speed and the training time vary. Hence, this work is based on the performance evaluation of the above classifiers, and a comparison of their performances with the existing MLH-based decoder. For the performance comparison, the CM is adopted, having entries in the form given in Table 8. The inference drawn from Table 7 is that the NB classifiers provide good decoding accuracies with lower training times, compared to the EBDT classifier. So, all three classifiers are selected for a fair comparison of their decoding performance with the MLH decoder. Further, the decoding accuracies are evaluated for the given classifiers for various values of SNR, until the accuracies increased to 100%, i.e., a perfect classification.

The validation CM for all the three classifiers is shown in Table 9, for a validation dataset of 4000 observations, for a given validation run. Table 9 is a combined representation of the performance of each classifier, shown with a different color; for example, the red-colored entries specify the corresponding values for the GNB classifier, and the green and blue-colored entries specify the obtained values for the KNB and EBDT classifiers, respectively. Further, the validation performances of the classifiers can be analyzed from Fig. 8, where Figs. 8(a), 8(b), 8(c), and 8(d), respectively represent the decoding accuracy, misclassifications cost, prediction speed, and the training time.

Another tabular structure, i.e., Table 10 is utilized for testing the classifier's performance, as shown in Section 3.3, and is described next. Lastly, the testing is performed 10 times for each SNR value, and the obtained results are shown in Table 11, where the red, green, blue, and brown color entries respectively represent the corresponding values of GNB, KNB, EBDT, and MLH.

3.3. Model testing

For testing the classifier's performance, 10 test datasets, each of 400 noisy QPSK symbols are generated for a given SNR having a uniform distribution over the classes. For the given range of SNRs, the performance metrics of test accuracy (correct classifications percentage) and the number of errors (misclassifications) are selected against the same metrics in MLH decoding. The simulation test performed with the given classifiers upon one of the test datasets generated earlier gives the resulting CM, as shown in Table 10.

4. Simulation results and discussion

The simulations are performed on a Windows 10 PC with a dual-core, Intel i-5 2.90 GHz processor, with 16 GB of installed RAM. The

Table 6
Classifier validation results for 25 runs (each @ 5 dB SNR).

SNR (dB)	Run #	GNB				KNB				EBDT			
		Accuracy (%)	Cost (Samples)	Prediction speed ~(obs/s)	Training time (s)	Accuracy (%)	Cost (Samples)	Prediction speed ~(obs/s)	Training time (s)	Accuracy (%)	Cost (Samples)	Prediction speed ~(obs/s)	Training time (s)
	1	93	282	27 000	6.5871	93	282	980	24.923	93.1	277	730	150.39
	2	93	278	6300	10.904	92.9	284	900	25.665	93	280	770	153.01
	3	93	279	15 000	10.201	93	282	900	29.054	93	278	1100	126.96
	4	93	279	8000	14.693	93.2	274	1200	30.995	93.2	274	540	160.48
	5	93	278	14 000	9.8813	93	278	840	33.618	93	279	840	152.85
	6	93	282	8700	29.655	93	282	560	47.272	93	280	690	184.31
	7	93	278	20 000	7.3765	92.9	284	1400	22.93	92.9	284	750	169.86
	8	93	281	11 000	9.251	92.9	283	1000	29.508	93	278	870	143.08
	9	93	282	18 000	9.6128	93	279	740	28.321	93	280	1000	151.61
	10	93	279	9300	8.6036	93.1	275	1100	33.266	92.8	290	750	148.51
	11	93	281	16 000	10.823	92.9	283	790	30.941	93	280	990	148.9
	12	93	282	7900	36.266	93	282	780	49.873	93.1	275	620	196.7
5	13	93	282	6200	16.303	93	279	860	38.77	93	280	1100	151.87
	14	93	278	12 000	8.4907	93	279	680	33.024	93.1	275	960	130.28
	15	93	281	9400	10.957	92.9	283	700	35.95	92.9	283	520	193.25
	16	93	282	21 000	12.003	93	279	500	43.716	93.1	275	1200	182.87
	17	93	282	6600	13.056	93	279	1000	28.023	92.9	284	670	179.17
	18	93	278	13 000	7.9711	92.9	284	820	24.34	93	280	980	143.2
	19	93	281	12 000	9.3114	92.9	283	1000	24.498	92.9	283	790	130.69
	20	93	282	24 000	10.386	93	279	790	23.554	93	280	1500	119.25
	21	93	279	13 000	8.9689	93.1	275	990	27.069	92.7	292	860	168.94
	22	93	278	21 000	7.6051	92.9	284	1200	20.381	92.9	284	840	127.2
	23	93	278	13 000	9.5657	93	279	980	33.02	93	279	680	149.46
	24	93	282	15 000	8.9842	93	279	800	27.12	93	280	870	125.73
	25	93	279	20 000	8.0119	93.1	275	900	29.256	92.8	290	1100	131.09

Table 7
The mean performance metrics (training and validation @ 5 dB SNR).

S.No.	Classifier	Validation accuracy (%)	Validation cost (samples)	Prediction speed ~(obs/s)	Training time (s)
1.	GNB	93	280.12	13 896	11.8187
2.	KNB	92.989	280.2	896.4	31.0035
3.	EBDT	92.976	280.8	868.8	152.786

MATLAB version 9.12.0.1884302 (R2022a), is used for the generation of datasets and performance assessment of the learning-based classifiers. The performance parameters are evaluated based on the CMs obtained with the experimentation. The CM for a given classifier is constructed using its class-wise performance results. The results comprise a tabular structure showing the correct decisions made by the classifier, as given in Table 8.

4.1. Performance measures

The performance parameters that are concerned with this paper include the *True Positives (TPs)*, *True Negatives (TNs)*, *False Positives (FPs)*, and *False Negatives (FNs)*, which can be analyzed with the CM. A CM is an $N \times N$ matrix, that represents the actual values and the predicted values in a matrix form, where N is the number of classes. The rows of a CM specify the *N true classes*, whereas the columns specify the *N predicted classes*.

Consider a 4×4 CM as shown in Table 8, where $a, b, c, \ldots, p$ represent the values within the respective cells of the matrix. The four performance parameters can be obtained from the given matrix as TP_n $(n = 0, 1, 2, 3)$ using the intersection of class n row and class n column, where the subscript represents the given class. Similarly, the FN_n is the sum of all the values of the corresponding row, except the TP_n. The FP_n value for a class is the sum of values of the corresponding column except for the TP_n value. The TN_n is determined by the sum of the values of all columns and rows except for class *n*. As an example, for class "0", we have TP_0 as *a*, FN_0 as *(b+c+d)*, FP_0 as *(e+i+m)*, and TN_0 is given as *(f+g+h+j+k+l+n+o+p)*. These parameters together specify the performance measures of a classifier concerning the given class (Eze et al., 2020):

- *Accuracy*: The ratio of correct predictions and total predictions, i.e.,

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

- *Error rate*: The ratio of incorrect predictions and total predictions, i.e.,

$$Error\ rate = 1 - Accuracy = \frac{FP + FN}{TP + TN + FN + FP} \tag{4}$$

- *Sensitivity/Recall/True Positive Rate (TPR)/Hit-Rate*: The ratio of correct positive predictions and total positives, i.e.,

$$Sensitivity\ (TPR) = \frac{TP}{TP + FN} \tag{5}$$

- *Specificity/True Negative Rate (TNR)*: The ratio of correct negative predictions and total negatives, i.e.,

$$Specificity\ (TNR) = \frac{TN}{TN + FP} \tag{6}$$

- *Precision/Positive Predictive Value (PPV)*: The ratio of correct positive predictions and total positive predictions, i.e.,

$$Precision\ (PPV) = \frac{TP}{TP + FP} \tag{7}$$

- *False Positive Rate (FPR)*: The ratio of incorrect positive predictions and total negatives, i.e.,

$$FPR = \frac{FP}{TN + FP} \tag{8}$$

- *Matthews Correlation Coefficient (MCC)* (Boughorbel et al., 2017): The correlation coefficient is calculated using all the four parameters of CM, i.e.,

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

- F_1 score: The harmonic mean of precision and sensitivity, i.e.,

$$F_1 = 2 \cdot \frac{Precision \cdot Sensitivity}{Precision + Sensitivity} \tag{10}$$

Table 8
A sample CM (4 × 4).

	Predicted class "0"	Predicted class "1"	Predicted class "2"	Predicted class "3"
True class "0"	*a*	*b*	*c*	*d*
True class "1"	*e*	*f*	*g*	*h*
True class "2"	*i*	*j*	*k*	*l*
True class "3"	*m*	*n*	*o*	*p*

Table 9
Classifier validation CM @ 5 dB SNR.

	Predicted class "0"			Predicted class "1"			Predicted class "2"			Predicted class "3"		
True class "0"	907	914	903	34	29	37	4	4	3	31	29	33
True class "1"	33	36	33	916	913	916	29	29	29	2	2	2
True class "2"	1	1	1	32	33	33	974	971	967	34	36	40
True class "3"	40	41	39	1	0	0	34	35	28	928	927	936
GNB				KNB			EBDT					

Table 10
Classifier testing CM @ 5 dB SNR.

	Predicted class "0"			Predicted class "1"			Predicted class "2"			Predicted class "3"		
True class "0"	99	99	100	5	5	4	0	0	0	3	3	3
True class "1"	1	1	1	87	87	88	4	4	3	0	0	0
True class "2"	0	0	0	4	4	4	97	97	97	4	4	4
True class "3"	1	1	1	0	0	0	8	7	5	87	88	90
GNB				KNB			EBDT					

Fig. 9 shows the ROC curves and the AUC, which are considered prominent metrics for the performance evaluation of the classifiers. The ROC is a probability curve that uses specificity and sensitivity measures and gives the TPR (i.e, sensitivity) for various thresholds of FPR (i.e., 1- specificity). The AUC provides a measure of separability between different classes, provided by the model. For a given class, the higher the value of AUC, the better the model predictions for that class. An AUC of 0.5 implies no class separability, whereas AUC values close to unity denote good separability between the classes. Further, the model predictions (correct/incorrect) with the NB and EBDT classifiers for a representative training/validation dataset are shown in Figs. 10 and 11, respectively, for an SNR of 5 dB. For each of the considered classifiers, the class-wise evaluation of the above performance measures and results is given in Table 12.

4.2. Result analysis and discussion

From the CM (Table 9), the total QPSK symbols generated for classes "0", "1", "2", and "3", are determined as $976, 980, 1041$, and 1003, respectively. This shows an approximate uniform distribution over the classes. The validation accuracy of a given classifier is determined by the CM. For example, 907 out of 976 $(907 + 34 + 4 + 31 = 976)$ class "0" symbols are positively predicted using GNB. This implies a TPR of $907/976 = 92.93\%$, for class "0". Also, the False Negative Rate (FNR) is seen to be $(34 + 4 + 31)/976 = 7.07\%$ for the same predicted class. Similarly, 936 symbols are predicted correctly out of 1003 $(936 + 39 + 0 + 28 = 1003)$ symbols of class "3" using EBDT. From Table 10, it can be observed that out of 107 symbols generated of class "0", 99 are correctly predicted (decoded) with the GNB and KNB classifiers, whereas 100 are correctly decoded with the EBDT classifier.

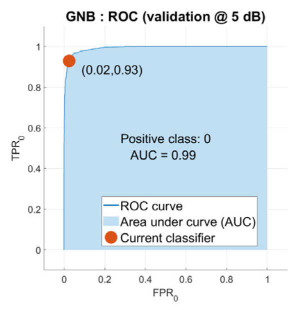
Using the performance measures mentioned in Section 4.1, the performances of all the three ML-based classifiers are tabulated in Table 12. The table shows that all the three ML-based classifiers have approximately the same performance metrics for the given validation dataset. Thus, the testing performance (average accuracy over all classes) of the NB decoders is almost the same (≈ 96.55%), and slightly higher than the EBDT decoder (96.52%). However, Table 7 shows significant differences between the training times of the classifiers. Similar to the testing performance, the NB classifiers have a validation accuracy of about 93% and a validation cost of about 280 samples. For the simulation scenario considered in this work, GNB offers the best performance with the highest prediction speed in observations/second (obs/s), of 13 896, followed by KNB (896.4), and EBDT (868.8), respectively. Similarly, the training times in seconds, required by the classifiers are seen as GNB (11.8187 s), KNB (31.0035 s), and EBDT (152.786 s), respectively. This shows improved performance of the GNB decoder than the KNB, and EBDT decoders.

On comparing the simulation test results of the MLH decoder with the GNB, KNB, and EBDT decoders, against the given SNR range, and using the results of Table 11, it is observed that all the three learning-based classifiers show comparable performance with the MLH decoder. The GNB and MLH decoders have almost the same median values for 10 test evaluations for each SNR value. Moreover, as the SNR increases, the KNB and the EBDT decoders also achieve the same performance as of GNB decoder. This can be seen in Fig. 12(a) for SNR values ≥ 9 dB for QPSK modulation. The circular dot and horizontal line segment inside the corresponding boxes represent the mean value and the median value of the prediction, respectively. Moreover, Fig. 12(b) shows the number of symbol errors in the decoding process. In both Figs. 12(a) and 12(b), as the SNR increases, the decoding accuracies also increase asymptotically towards 100%, and simultaneously the incurred symbol errors decrease asymptotically towards 0. At an SNR $\geq$ 11 dB, the decoding accuracy and the symbol errors are seen to converge at 100% and 0, respectively. This is also evident from the reduced Inter Quartile Range (IQR), shown in Figs. 12(a) and 12(b). Specifically, from Figs. 12(a) and 12(b), it is observed that as the SNR increases, the IQR approaches zero, and the mean and median converge. This shows that the ML-based decoders have performances similar to the MLH decoder. Moreover, the proposed decoders are reconfigurable because of their learning abilities.

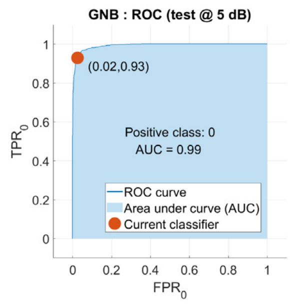
The performance comparison between the ML-based classifiers suggests the application of the GNB classifier as it requires the lowest training time, highest prediction speed, higher accuracy, and a moderate misclassification cost. The KNB and EBDT classifiers have similar misclassification costs and accuracies but differ in the prediction speed and training time, from the GNB classifier. The EBDT classifier requires

Table 11
Comparison of classifiers' test results.

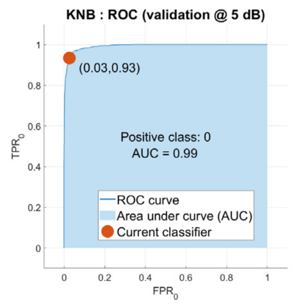
SNR (dB)	Test #	GNB		KNB		EBDT		MLH		SNR (dB)	Test #	GNB		KNB		EBDT		MLH	
		Accuracy (%)	Symbol errors	Accuracy (%)	Symbol errors	Accuracy (%)	Symbol errors	Accuracy (%)	Symbol errors			Accuracy (%)	Symbol errors	Accuracy (%)	Symbol errors	Accuracy (%)	Symbol errors	Accuracy (%)	Symbol errors
5	1	93	28	92.5	30	91.8	33	93.25	27	9	1	99.2	3	99.2	3	99.2	3	99.25	3
	2	93.2	27	93	28	93	28	93.5	26		2	99.2	3	99.2	3	99.5	2	99.25	3
	3	91.8	33	91.5	34	91.2	35	91.75	33		3	99.8	1	99.8	1	99.8	1	99.75	1
	4	94	24	94	24	94	24	94	24		4	100	0	100	0	100	0	100	0
	5	93.2	27	93.2	27	93.5	26	93.25	27		5	99.2	3	99.2	3	99.5	2	99.25	3
	6	92.2	31	92.2	31	92.2	31	92.5	30		6	100	0	100	0	99.8	1	99.75	1
	7	92.2	31	92.5	30	91.8	33	92.5	30		7	99.5	2	99.2	3	99	4	99.5	2
	8	94.5	22	94.5	22	93.5	26	94.5	22		8	99	4	99.2	3	99	4	99	4
	9	92.5	30	92	32	92.2	31	92.25	31		9	99.8	1	99.8	1	99.8	1	99.75	1
	10	92	32	92	32	92.2	31	91.5	34		10	99.5	2	99.5	2	99.5	2	99.5	2
6	1	95.8	17	96	16	96.2	15	96.25	15	10	1	100	0	100	0	100	0	100	0
	2	95.2	19	95.2	19	95.5	18	95.25	19		2	100	0	100	0	100	0	100	0
	3	95.5	18	96	16	96	16	95.25	19		3	99.8	1	99.8	1	99.8	1	99.75	1
	4	97	12	96.2	15	96.5	14	97	12		4	100	0	100	0	99.8	1	100	0
	5	96	16	95.5	18	95	20	95.75	17		5	100	0	100	0	100	0	100	0
	6	95.5	18	95.5	18	95.8	17	95.75	17		6	100	0	100	0	100	0	100	0
	7	95.2	19	95.2	19	95.5	18	95.25	19		7	100	0	100	0	100	0	100	0
	8	96.8	13	96.8	13	95.2	19	96.5	14		8	99.8	1	100	0	100	0	99.75	1
	9	96.5	14	96.5	14	95.8	17	96.75	13		9	99.8	1	99.8	1	99.8	1	99.75	1
	10	95.8	17	95	20	95.2	19	95.75	17		10	99.8	1	99.8	1	99.8	1	99.75	1
7	1	97.2	11	97.5	10	97	12	97.5	10	11	1	100	0	100	0	100	0	100	0
	2	96.8	13	96.8	13	96.8	13	96.75	13		2	100	0	100	0	100	0	100	0
	3	97.2	11	97.2	11	97.2	11	97.25	11		3	100	0	100	0	100	0	100	0
	4	97.8	9	97.8	9	97.5	10	97.75	9		4	100	0	100	0	100	0	100	0
	5	97.5	10	97.5	10	97.2	11	97.75	9		5	100	0	100	0	100	0	100	0
	6	96.8	13	97	12	97.5	10	97	12		6	100	0	100	0	100	0	100	0
	7	96.5	14	96.8	13	96.2	15	96.75	13		7	100	0	100	0	100	0	100	0
	8	96.5	14	96.2	15	95.8	17	96.25	15		8	100	0	100	0	100	0	100	0
	9	97.2	11	97.2	11	97.2	11	97.25	11		9	100	0	100	0	100	0	100	0
	10	98	8	97.8	9	97.8	9	97.75	9		10	100	0	99.8	1	99.8	1	100	0
8	1	99.2	3	99	4	98.2	7	99.25	3	12	1	100	0	100	0	100	0	100	0
	2	99.2	3	99.5	2	99	4	99.25	3		2	100	0	100	0	100	0	100	0
	3	99	4	99	4	98.5	6	98.75	5		3	100	0	100	0	100	0	100	0
	4	99.2	3	99.2	3	99.2	3	99.25	3		4	100	0	100	0	100	0	100	0
	5	98.5	6	98.8	5	99	4	98.25	7		5	100	0	100	0	100	0	100	0
	6	99.2	3	99.2	3	98.8	5	99.25	3		6	100	0	100	0	100	0	100	0
	7	98.5	6	98.5	6	98	8	98.5	6		7	100	0	100	0	100	0	100	0
	8	98.8	5	98.8	5	99	4	98.75	5		8	100	0	100	0	100	0	100	0
	9	99.2	3	99.2	3	99.2	3	99.25	3		9	100	0	100	0	100	0	100	0
	10	99	4	99	4	99	4	99	4		10	100	0	100	0	100	0	100	0



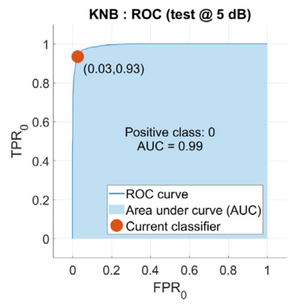
(a) Area Under the Curve (AUC)/Receiver Operating Characteristics (ROC) for GNB (validation)



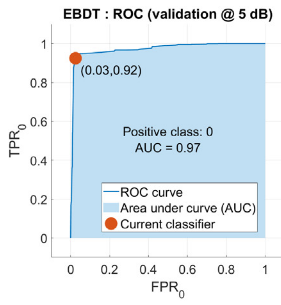
(b) AUC/ROC for GNB (test)



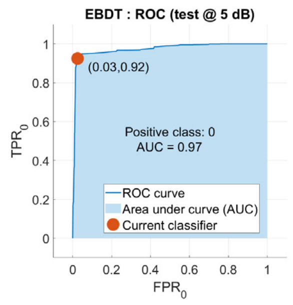
(c) AUC/ROC for KNB (validation)



(d) AUC/ROC for KNB (test)



(e) AUC/ROC for EBDT (validation)



(f) AUC/ROC for EBDT (test)

Fig. 9. ROC and AUC for GNB, KNB and EBDT classifiers.

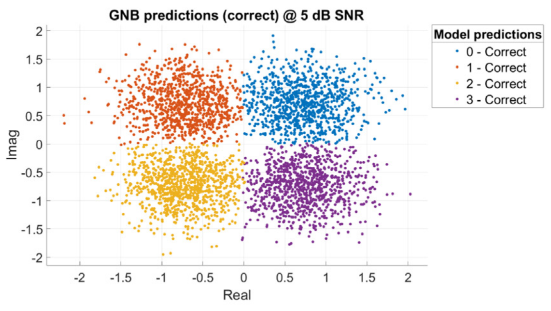
Table 12
Performance metrics for classifiers (training/validation @ 5 dB SNR).

Performance metric	GNB predictor classes				KNB predictor classes				EBDT predictor classes			
	0	1	2	3	0	1	2	3	0	1	2	3
Accuracy (%)	96.425	96.677	96.650	96.450	96.500	96.753	96.579	96.425	96.350	96.650	96.650	96.450
Error rate (%)	3.575	3.323	3.350	3.550	3.500	3.247	3.421	3.575	3.650	3.350	3.350	3.550
Sensitivity (%)	92.930	93.470	93.560	92.520	93.648	93.163	93.276	92.423	92.520	93.469	92.891	93.320
Specificity (%)	97.550	97.710	97.730	97.760	97.421	97.928	97.728	97.764	97.586	97.682	97.972	97.497
hline Precision (%)	92.450	92.990	93.560	93.260	92.137	93.641	93.455	93.259	92.520	92.901	94.158	92.581
FPR	0.024	0.023	0.023	0.022	0.026	0.021	0.023	0.022	0.024	0.023	0.020	0.025
MCC	0.903	0.910	0.913	0.905	0.997	0.912	0.910	0.905	0.901	0.910	0.913	0.906
F_1 score	92.689	93.229	93.560	92.889	92.886	93.401	93.365	92.839	92.520	93.184	93.520	92.949

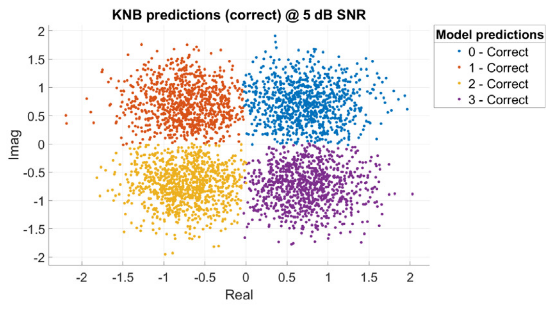
the highest training time (152.786 s) out of all the three classifiers. This can also be seen in Table 7.

Similarly, Fig. 9 shows the ROC curves (for class "0") obtained after the validation and testing of the considered learning-based classifiers at 5 dB SNR. The ROC curves are plotted considering their performance for class "0"; hence the horizontal axis refers to the FPR and the vertical axis refers to the TPR for class "0" symbols. These results confirm that the same AUCs are obtained during both training and validation. Moreover, the NB classifiers are preferable to the EBDT classifier due to their slightly higher AUC values. The high AUC values, and the performance metrics in Table 12, together indicate an admirable learning-based model performance.

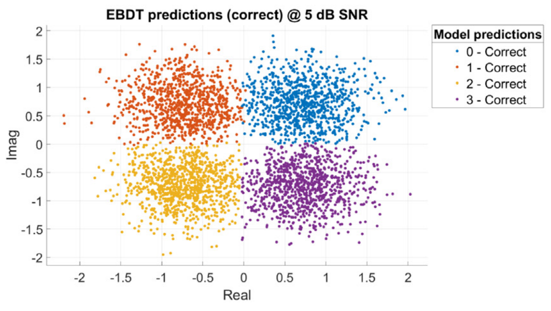
Moreover, the results of Table 12 are computed considering a low SNR scenario. As the modern and forthcoming communication systems



(a) Correct predictions (GNB)

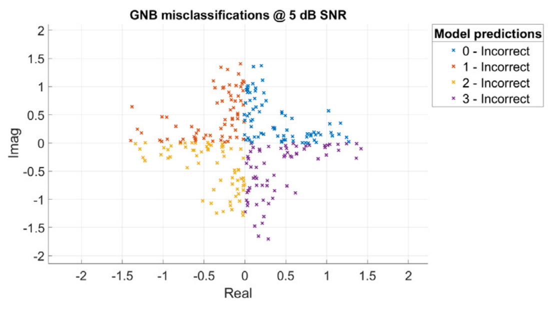


(b) Correct predictions (KNB)

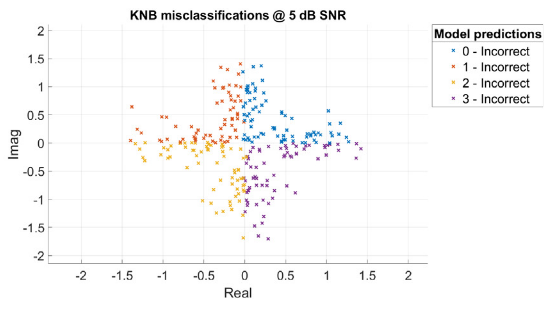


(c) Correct predictions (EBDT)

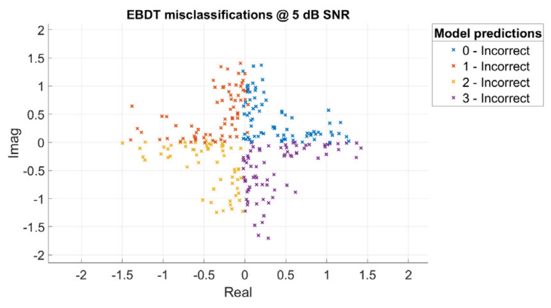
Fig. 10. Correct class predictions with learning-based classifiers (SNR = 5 dB).



(a) Incorrect predictions (GNB)



(b) Incorrect predictions (KNB)

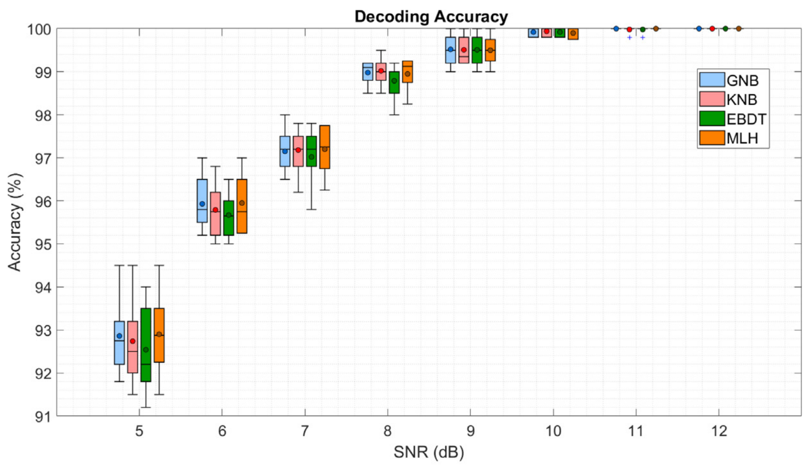


(c) Incorrect predictions (EBDT)

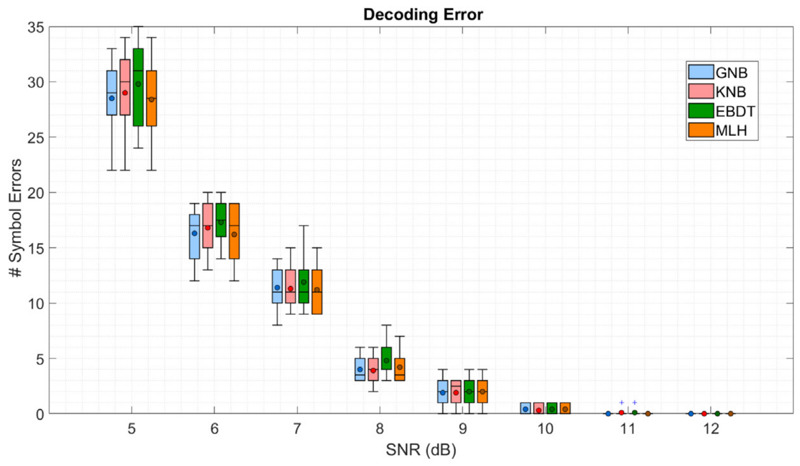
Fig. 11. Misclassifications with learning-based classifiers (SNR = 5 dB).

shall deliver a much higher SNR, very high accuracies can be obtained with our model. It can be seen from Figs. 12 and 13, that the same ML-based model can be trained with different modulation schemes and it provides a comparable result with the already existing MLH decoder. However, the latter has no option for reconfigurability.

The reconfigurable nature of the learning-based decoder can be visualized with the performance of the proposed model for the BPSK constellation diagram, as shown in Figs. 13(a) and 13(b). For the system employing BPSK modulation, new datasets were generated similar to the QPSK-MLbD datasets, but having only two classes ("0" and "1"). Correspondingly, the number of samples in the training/validation datasets and testing datasets, are kept as 2000 and 200, respectively. The model is trained with the new (BPSK-MLbD) datasets, and the decoder performance is plotted in Figs. 13(a) and 13(b). For the BPSK



(a) Decoding accuracy performance comparison



(b) Decoding error performance comparison

Fig. 12. Performance comparison of proposed decoders with QPSK constellation.

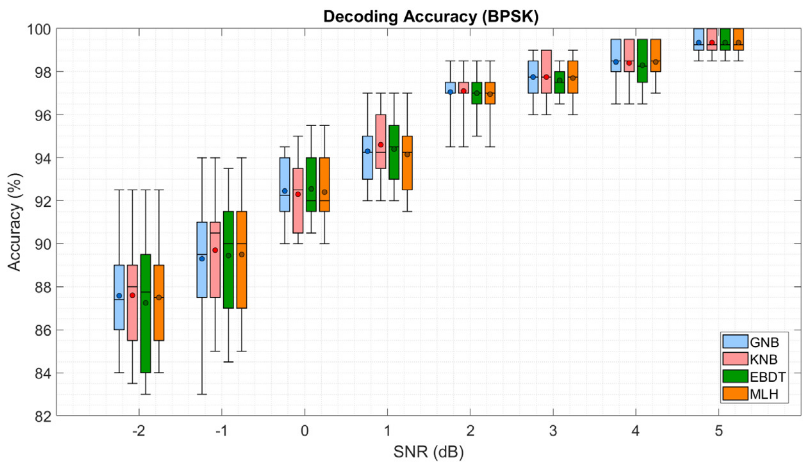
system also, the performance of the existing MLH decoder is comparable with that of the proposed ML-based decoders, validating the replacement of the MLH decoder with the ML-based decoder, in an SNR range of −2 to 5 dB.

For QPSK modulation, Table 12 shows high decoding accuracy values (success rates) for all the considered classifiers. For example, each of the classifiers has an accuracy of about 96.5% over all the four classes, at an SNR of 5 dB. Similarly, the MCC score for each of the classifiers is nearly 93. Also, there is a little variation among the class performances of each of the classifiers. This is quite natural, as the training samples themselves have a uniformly random distribution. The accuracy of the MLH decoder over the test dataset gives similar results. However, implementing the decoder through a learning-based approach, such as ours, makes the model highly reconfigurable, and suitable for next-generation systems.

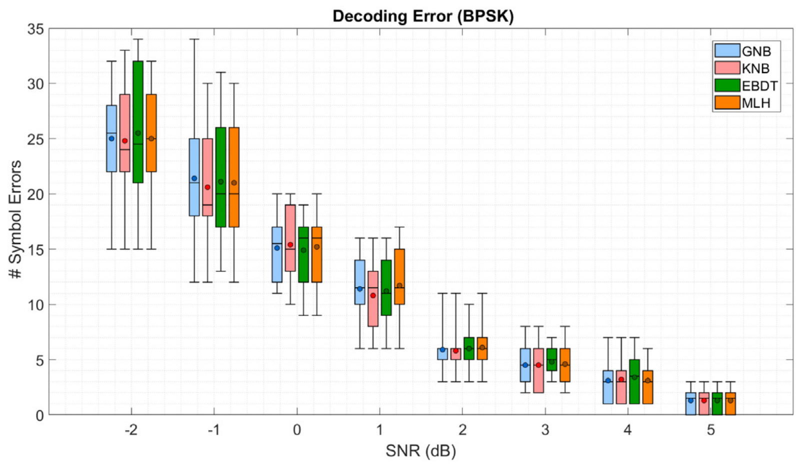
The proposed decoder can be utilized for decoding both users' signals in a two-user PD-NOMA network, simultaneously, without the requirement of a SIC, as indicated in Section 2. Conventionally, the SIC block in NOMA is used to decode each user's message (in a sequential manner). For the user's signals decoding via ML-based decoders in the NOMA system, the SIC block is eliminated, as no cancellation and re-transmission of signals are required. This can be done by allocating different fixed power levels to both the NOMA users, determining the superposed constellation point, and then resolving the superposed points into the original constellation symbols, as indicated in Fig. 2(b). However, the exact algorithm is a part of our future work and is in the development stage. The real-time datasets for application to the two-user PD-NOMA system, are also being generated as an extension of the current work. It is expected that the proposed model also reduces the corresponding error probability of the system, due to the absence of the SIC block in the NOMA receiver. Further, latency reduction will also be observed in such systems, due to the elimination of the SIC.

5. Conclusions

In this work, the application of an ML-based symbol decoder is proposed as a replacement to the traditional MLH decoder in a QPSK system, where both the decoders have comparable performance, but the former is reconfigurable. Since the model is based on the ML approach rather than the DL approach, it has lower complexity indicating a faster response. The model utilizes only the received symbols to extract all the required predictors. Appropriate low-complexity datasets are also generated utilizing the received symbols. After investigating the performance of nineteen standard ML classifiers, three well-known classifiers, viz: the KNB, the GNB, and the EBDT are found to be appropriate for the proposed system, for both QPSK and BPSK modulations. Additionally, the model shows a high decoding accuracy and low decoding error even for small SNR values. Overall, the test results show the appropriateness of the GNB classifier for the proposed model. The learning property, and the dataset generation mechanism together make the proposed model invariant of channel fading, thereby eliminating the complex channel estimation strategies. Further, the model supports the MIMO configuration and can simultaneously decode PD-NOMA users reducing



(a) Decoding accuracy performance comparison



(b) Decoding error performance comparison

Fig. 13. Performance comparison of proposed decoders with BPSK constellation.

network latency in next-generation systems where it also solves the problem of SIC.

CRediT authorship contribution statement

Saurabh Srivastava: Conceptualization, Visualization, Investigation, Methodology, Model development, Data-set preparation, Validation, Writing – original draft. **Prajna Parimita Dash:** Conceptualization, Model review, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The following datasets that support the findings of this study are available from the corresponding author upon reasonable request.

QPSK-MLbD Datasets: The dataset for QPSK symbol decoding, for use with the ML-based classifiers.

BPSK-MLbD Datasets: The dataset for BPSK symbol decoding, for use with the ML-based classifiers.

Acknowledgments

The authors would like to acknowledge the support of the Department of Electronics and Communication Engineering, Birla Institute of Technology, Mesra, India, in providing the necessary resources for the completion of this work.

References

Bari, M., Taher, H., Sherazi, S.S., Doroslovacki, M., 2016. Supervised machine learning for signals having rrc shaped pulses. In: 2016 50th Asilomar Conf. Sig. Sys. Comp.. IEEE, pp. 652–656. http://dx.doi.org/10.1109/ACSSC.20167869124.

Bhavani, A., Kumar, B.S., 2021. A review of state art of text classification algorithms. In: IEEE 2021 5th Int. Conf. Comput. Method. Commun. (ICCMC). IEEE, pp. 1484–1490. http://dx.doi.org/10.1109/ICCMC5101920219418262.

Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., Sillero-Denamiel, M.R., 2021. Variable selection for naïve bayes classification. Comput. Oper. Res. 135, 105456. http://dx.doi.org/10.1016/j.cor.2021105456.

Boughorbel, S., Jarray, F., El-Anbari, M., 2017. Optimal classifier for imbalanced data using matthews correlation coefficient metric. PLoS One 12 (6), e0177678. http://dx.doi.org/10.1371/journal.pone.0177678.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140. http://dx.doi.org/10.1007/BF00058655.

Charalampakis, B., Spathis, D., Kouslis, E., Kermanidis, K., 2016. A comparison between semi-supervised and supervised text mining techniques on detecting irony in greek political tweets. Eng. Appl. Artif. Intell. 51, 50–57. http://dx.doi.org/10.1016/j.engappai.201601007.

Chen, H., Niu, W., Zhao, Y., Zhang, J., Chi, N., Li, Z., 2021. Adaptive deep-learning equalizer based on constellation partitioning scheme with reduced computational complexity in uvlc system. Opt. Express 29 (14), 21773–21782. http://dx.doi.org/10.1364/OE.432351.

Drissi, M., Oumsis, M., Aboutajdine, D., 2017. A multi-criteria decision framework for network selection over lte and wlan. Eng. Appl. Arti. Intell. 66, 113–127. http://dx.doi.org/10.1016/j.engappai.201708019.

Eze, P., Parampalli, U., Evans, R., Liu, D., 2020. Evaluation of the effect of steganography on medical image classification accuracy. J. Appl. Bioinform. Comput. Biol. 9 (4), 2.

Garnier, J.-R., Fabre, A., Farès, H., Bonnefoi, R., 2020. On the performance of qpsk modulation over downlink noma: From error probability derivation to sdr-based validation. IEEE Access 8, 66495–66507. http://dx.doi.org/10.1109/ACCESS.20202983299.

Ghosh, S., Bandyopadhyay, A., Sahay, S., Ghosh, R., Kundu, I., Santosh, K., 2021. Colorectal histology tumor detection using ensemble deep neural network. Eng. Appl. Artif. Intell. 100, 104202. http://dx.doi.org/10.1016/j.engappai.2021104202.

Hillebrand, E., Lukas, M., Wei, W., 2021. Bagging weak predictors. Int. J. Forecast. 37 (1), 237–254. http://dx.doi.org/10.1016/j.ijforecast.202005002.

Kotsiantis, S.B., 2014. Bagging and boosting variants for handling classifications problems: a survey. Knowl. Eng. Rev. 29 (1), 78–100. http://dx.doi.org/10.1017/S0269888913000313.

Kotsiantis, S.B., Zaharakis, I., Pintelas, P., et al., 2007. Supervised machine learning: A review of classification techniques. Emerg. Artif. Intell. Appl. Comput. Eng. 160 (1), 3–24, https://www.informatica.si/index.php/informatica/article/download/148/140.

Kukreja, V., et al., 2021. A retrospective study on handwritten mathematical symbols and expressions: Classification and recognition. Eng. Appl. Artif. Intell. 103, 104292. http://dx.doi.org/10.1016/j.engappai.2021104292.

Kupervasser, O., 2014. The mysterious optimality of naive bayes: Estimation of the probability in the system of classifiers. Pattern. Recognit. Im. Anal. 24 (1), 1–10. http://dx.doi.org/10.1134/S1054661814010088.

Li, Y., Carabelli, S., Fadda, E., Manerba, D., Tadei, R., Terzo, O., 2020. Machine learning and optimization for production rescheduling in industry 40. Int. J. Adv. Manuf. Technol. 110 (9), 2445–2463. http://dx.doi.org/10.1007/s00170-020-05850-5.

Liao, F., Wei, S., Zou, S., 2020. Deep learning methods in communication systems: A review. J. Phys.: Conf. Ser. 1617, 012024. http://dx.doi.org/10.1088/1742-6596/1617/1/012024.

Liu, H., Ziping, W., Zhang, H., Li, B., Zhao, C., 2022. Tiny machine learning (tiny-ml) for efficient channel estimation and signal detection. IEEE Trans. Veh. Technol. 1. http://dx.doi.org/10.1109/TVT.20223163786.

Lyke, J.C., Christodoulou, C.G., Vera, G.A., Edwards, A.H., 2015. An introduction to reconfigurable systems. Proc. IEEE 103 (3), 291–317. http://dx.doi.org/10.1109/JPROC.20152397832.

Matta, M., Cardarilli, G.C., Nunzio, L.Di., Fazzolari, R., Giardino, D., Nannarelli, A., Re, M., Spano, S., 2019. A reinforcement learning-based qam/psk symbol synchronizer. IEEE Access 7, 124147–124157. http://dx.doi.org/10.1109/ACCESS.20192938390.

Men, H., Jin, M., 2014. A low-complexity ml detection algorithm for spatial modulation systems with m psk constellation. IEEE Commun. Lett. 18 (8), 1375–1378. http://dx.doi.org/10.1109/LCOMM.20142331283.

Meng, L., Su, H., Lou, C., Li, J., 2022. Cross-domain mutual information adversarial maximization. Eng. Appl. Artif. Intell. 110, 104665. http://dx.doi.org/10.1016/j.engappai.2022104665.

Moon, S.-R., Kim, E.-S., Sung, M., Rha, H.Y., Lee, E.S., Lee, I.-M., Park, K.H., Lee, J.K., Cho, S.-H., 2022. 6 g indoor network enabled by photonics-and electronics-based sub-thz technology. J. Light Technol. 40 (2), 499–510. http://dx.doi.org/10.1109/JLT.20213113898.

Morocho-Cayamcela, M.E., Lee, H., Lim, W., 2019. Machine learning for 5g/b5 g mobile and wireless communications: Potential, limitations, and future directions. IEEE Access 7, 137184–137206. http://dx.doi.org/10.1109/ACCESS.20192942390.

Nauryzbayev, G., Omarov, O., Arzykulov, S., Rabie, K.M., Li, X., Eltawil, A.M., 2021. Performance limits of wireless powered cooperative noma over generalized fading. Trans. Emerg. Telecommun. Technol. http://dx.doi.org/10.1002/ett.4415.

Njoku, O.C., 2019. Decision trees and their application for classification and regression problems. https://bearworks.missouristate.edu/cgi/viewcontent.cgi?article=4409&context=theses.

Norolahi, J., Azmi, P., 2021. Blind modulation classification via combined machine learning and signal feature extraction. doi:https://arxiv.org/abs/210104337.

Ostad-Ali-Askari, K., Shayannejad, M., Ghorbanizadeh-Kharazi, H., 2017. Artificial neural network for modeling nitrate pollution of groundwater in marginal area of zayandeh-rood river, isfahan, iran. KSCE J. Civ. Eng. 21 (1), 134–140. http://dx.doi.org/10.1007/s12205-016-0572-8.

Park, D.J., Park, M.W., Lee, H., Kim, Y.-J., Kim, Y., Park, Y.H., 2021. Development of machine learning model for diagnostic disease prediction based on laboratory tests. Sci. Rep. 11 (7567), 1–11. http://dx.doi.org/10.1038/s41598-021-87171-5.

Pauluzzi, D.R., Beaulieu, N.C., 2000. A comparison of snr estimation techniques for the awgn channel. IEEE Trans. Commun. 48 (10), 1681–1691. http://dx.doi.org/10.1109/26871393.

Pei, X., Chen, Y., Wen, M., Yu, H., Panayirci, E., Poor, H.V., 2022. Next-generation multiple access based on noma with power level modulation. IEEE J. Sel. Area. Commun. 40 (4), 1072–1083. http://dx.doi.org/10.1109/JSAC.20223143240.

Peng, S., Jiang, H., Wang, H., Alwageed, H., Yao, Y.-D., 2017. Modulation classification using convolutional neural network based deep learning model. In: 2017 26th Wire. Opt. Commun. Conf. (WOCC). IEEE, pp. 1–5. http://dx.doi.org/10.1109/WOCC.20177929000.

Peng, S., Sun, S., Yao, Y.-D., 2021. A survey of modulation classification using deep learning: Signal representation and data preprocessing. IEEE Trans. Neural Netw. Learn. Syst. http://dx.doi.org/10.1109/TNNLS.20213085433.

Qureshi, A.R., Nasir, A.A., Masood, M., Abdallah, S., 2021. Resource allocation for a multi-device urllc network. Trans. Emerg. Telecommun. Technol. http://dx.doi.org/10.1002/ett.4424.

Saif, W.S., Esmail, M.A., Ragheb, A.M., Alshawi, T.A., Alshebeili, S.A., 2020. Machine learning techniques for optical performance monitoring and modulation format identification: A survey. IEEE Commun. Surv. Tutor. 22 (4), 2839–2882. http://dx.doi.org/10.1109/COMST.20203018494.

Shen, S., Lu, H., Sadoughi, M., Hu, C., Nemani, V., Thelen, A., Webster, K., Darr, M., Sidon, J., Kenny, S., 2021. A physics-informed deep learning approach for bearing fault detection. Eng. Appl. Artif. Intell. 103, 104295. http://dx.doi.org/10.1016/j.engappai.2021104295.

Simeone, O., 2018. A very brief introduction to machine learning with applications to communication systems. IEEE Trans. Cogn. Commun. Netw. 4 (4), 648–664. http://dx.doi.org/10.1109/TCCN.20182881442.

Soula, A., Tbarki, K., Ksantini, R., Said, S.B., Lachiri, Z., 2020. A novel incremental kernel nonparametric svm model (ikn-svm) for data classification: An application to face detection. Eng. Appl. Artif. Intell. 89, 103468. http://dx.doi.org/10.1016/j.engappai.2019103468.

Sparjan, R.J., Thirunavukkarasu, M., Thangavelu, L., 2021. Reconfigurable intelligent surface aided coordinated multipoint transmission for the simultaneous wireless information and power transfer non-orthogonal multiple access network. Trans. Emerg. Telecommun. Technol. http://dx.doi.org/10.1002/ett.4434.

Sun, H., Zhang, Y., Wang, F., Zhang, J., Shi, S., 2021. Svm aided signal detection in generalized spatial modulation vlc system. IEEE Access 9, 80360–80372. http://dx.doi.org/10.1109/ACCESS.20213084823.

Tan, Y., Zhao, F., He, M., Wang, Y., Zhou, W., Zhang, J., Zhu, M., Shi, Y., Yu, J., 2022. Transmission of high-frequency terahertz band signal beyond 300 ghz over metallic hollow core fiber. J. Light Technol. 40 (3), 700–707. http://dx.doi.org/10.1109/JLT.20213123473.

Tg, E., 2018. Etsi ts 138 211 v1520 (2018-7) Technical specifications. https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/150200_60/ts_138211V150200p.pdf.

Toğaçar, M., 2022. Detecting attacks on iot devices with probabilistic bayesian neural networks and hunger games search optimization approaches. Trans. Emerg. Telecommun. Technol. e4418. http://dx.doi.org/10.1002/ett.4418.

Vairagade, R.S., SH, B., 2021. Enabling machine learning-based side-chaining for improving qos in blockchain-powered iot networks. Trans. Emerg. Telecommun. Technol. 33 (4), e4433. http://dx.doi.org/10.1002/ett.4433.

Wang, P., Xu, Z., 2020. A novel consumer purchase behavior recognition method using ensemble learning algorithm. Math. Probl. Eng. 2020 (10), http://dx.doi.org/10.1155/2020/6673535.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., et al., 2008. Top 10 algorithms in data mining. Knowl. Inf. Syst. 14 (1), 1–37. http://dx.doi.org/10.1007/s10115-007-0114-2.

Xie, X., Peng, S., Yang, X., 2020. Deep learning-based signal-to-noise ratio estimation using constellation diagrams. Mob. Inf. Syst. 2020, 9. http://dx.doi.org/10.1155/2020/8840340.

Zhang, H., 2005. Exploring conditions for the optimality of naïve bayes. Int. J. Pattern Recognit. Artif. Intell. 19 (2), 183–198. http://dx.doi.org/10.1142/S0218001405003983.