Published *online* on the journal's web page : http://innovatics.unsil.ac.id

Innovation in Research of Informatics (INNOVATICS)

| ISSN (Online) 2656-8993 |

Approximate Bayesian Inference for Confidence-Aware DNA Sequence Classification Using Monte Carlo Dropout

Nur Alamsyah[1], Budiman[2], Reni Nursyanti[3], Elia Setiana[4], Venia Restreva Danestiara[5]

[1]*Information System,Universitas Informatika Dan Bisnis Indonesia, Jl Soekarno Hatta, Bandung 40285, Indonesia*
[1]*nuralamsyahmail@unibi.ac.id,* [2]*budiman@unibi.ac.id,* [3]*reninursyanti@unibi.ac.id,* [4]*elia.setiana@unibi.ac.id,* [5]*veniarestreva@unibi.ac.id*

ARTICLE INFORMATION

Article History:
Received: January 28, 2025
Last Revision: April 16, 2025
Published *Online*: April 30, 2025

KEYWORDS

Monte Carlo Dropout,
Bayesian Confidence Quantification,
DNA Sequence Classification,
Splice Junction Detection,
Uncertainty Estimation

CORRESPONDENCE

Phone: 08122301680
E-mail: nuralamsyah@unibi.ac.id

ABSTRACT

Splice junction classification in DNA sequences is critical for understanding genetic structures, particularly in identifying exon-intron (EI), intron-exon (IE), and neither boundary. Traditional neural networks achieve high accuracy but lack the ability to quantify uncertainty an essential aspect in bioinformatics. In this study, we propose a method for confidence-aware DNA sequence classification by applying Approximate Bayesian Inference using Monte Carlo Dropout (MCD). We conducted experiments on a publicly available dataset comprising 3,187 DNA sequences, each encoded with 180 binary features. A baseline neural network achieved a test accuracy of 95.61%, while the proposed MCD-enhanced model improved performance to 96.03% and simultaneously provided uncertainty estimates through multiple inference sampling. The uncertainty analysis enabled the identification of low-confidence predictions, improving model interpretability and reliability. This research contributes a practical approach that balances accuracy and uncertainty estimation, making it suitable for critical genomic applications requiring robust and explainable predictions.

1. INTRODUCTION

Splice junction classification in DNA sequences is a critical task in bioinformatics, enabling the identification of exon-intron (EI) and intron-exon (IE) boundaries within genetic structures [1]. Accurate detection of these splice junctions is essential for understanding gene expression, protein synthesis, and genetic variations [2]. Traditional methods for solving this problem often involve rule-based systems or classical machine learning approaches, such as Support Vector Machines (SVM) and Random Forest, which rely heavily on handcrafted features and domain expertise [3]. While these methods achieve reasonable accuracy, they lack scalability and struggle to generalize to unseen data due to their reliance on fixed feature representations [4].

Previous research on splice junction classification in DNA sequences has employed various traditional and machine learning-based methods [5]. Rule-based systems were among the earliest approaches, leveraging predefined biological patterns to identify splice junctions [6]. While straightforward, these systems often suffered from limited generalizability and reliance on expert-defined heuristics [7]. Classical machine learning models, such as Support Vector Machines (SVM), Random Forest, and Hidden Markov Models (HMM), have shown improvements by utilizing numerical feature representations and statistical patterns within DNA sequences [8]. These methods, however, depend heavily on handcrafted features, such as nucleotide frequencies or position-specific scoring matrices, which are challenging to optimize for large-scale datasets or complex sequences.

Deep learning techniques have recently gained traction in DNA sequence analysis due to their ability to learn features automatically from raw data without manual feature engineering [9]. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been particularly effective for capturing spatial and sequential patterns in genomic data [10]. Studies employing these architectures have demonstrated state-of-the-art accuracy in sequence classification tasks. However, most deep learning models are deterministic, providing point predictions without any measure of confidence or uncertainty [11]. This limitation is critical in

bioinformatics applications, where decisions often need to be supported by robust reliability metrics.

To address these issues, Bayesian approaches have been explored for modeling uncertainty in neural networks. Bayesian Neural Networks (BNNs) extend standard neural networks by modeling weights as probability distributions, allowing for Bayesian inference over predictions. Despite their theoretical strengths, the computational cost of training BNNs and the complexity of posterior approximation have hindered their adoption in large-scale genomic studies [12]. Monte Carlo Dropout (MCD) emerges as an efficient alternative, approximating Bayesian inference by retaining dropout layers during inference to estimate predictive uncertainty [13]. This approach balances computational efficiency and the ability to quantify uncertainty, making it well-suited for applications in DNA sequence classification.

In this study, we propose the use of Monte Carlo Dropout to incorporate approximate Bayesian inference for confidence quantification in DNA sequence classification. The research aims to evaluate the effectiveness of MCD in improving classification performance while providing uncertainty estimation. By comparing a baseline neural network with the MCD-enhanced model, we highlight the advantages of integrating Bayesian confidence quantification into sequence classification tasks. The results demonstrate that MCD not only enhances prediction accuracy but also offers actionable insights into model reliability, addressing a critical need in bioinformatics research.

2. RELATED WORK

Splice junction classification in DNA sequences has been extensively studied due to its importance in understanding genetic mechanisms and guiding research in genomics. Early approaches relied heavily on rule-based systems, which used predefined biological patterns to identify splice junctions. For example, Oh *et al*. [14] proposed a system that matched specific nucleotide sequences to identify exon-intron (EI) and intron-exon (IE) boundaries. Although effective for simple cases, these methods lacked scalability and robustness when applied to large-scale genomic datasets.

With the advent of machine learning, researchers began employing statistical models to improve the accuracy and generalizability of splice junction classification. Hidden Markov Models (HMMs) were among the first probabilistic approaches used in this domain [15]. HMMs captured the sequential nature of DNA sequences but required manual feature extraction, limiting their ability to adapt to complex patterns. Similarly, Support Vector Machines (SVMs) and Random Forests [16] achieved reasonable performance by leveraging handcrafted features such as nucleotide frequencies and position-specific scoring matrices. However, these approaches were constrained by their reliance on feature engineering and their inability to generalize to diverse datasets.

Deep learning has revolutionized genomic research by enabling models to learn directly from raw data without manual feature engineering. Convolutional Neural Networks (CNNs) have been successfully applied to DNA sequence classification due to their ability to extract spatial features from nucleotide encodings [17]. Similarly, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have shown promise in capturing sequential dependencies within DNA sequences [18]. Despite achieving state-of-the-art accuracy, these models are deterministic, providing no measure of uncertainty in their predictions, which limits their interpretability and reliability in critical applications.

To address the limitations of deterministic models, Bayesian Neural Networks (BNNs) have been explored as a means of incorporating uncertainty quantification into neural networks [19]. BNNs model weights as probability distributions, allowing for Bayesian inference over predictions. However, the computational complexity of training BNNs and the challenges in posterior approximation have hindered their widespread adoption. To overcome these limitations, Gal and Ghahramani [8] introduced Monte Carlo Dropout (MCD), an efficient approximation to Bayesian inference. By retaining dropout layers during inference, MCD enables models to estimate predictive uncertainty while maintaining computational efficiency. This approach has been successfully applied to various domains, including computer vision and natural language processing, but its application in bioinformatics, particularly for DNA sequence classification, remains underexplored.

In this study, we build on the strengths of Monte Carlo Dropout by applying it to DNA splice junction classification. The goal is to evaluate the effectiveness of MCD in not only improving classification accuracy but also providing actionable uncertainty estimates for each prediction. By addressing the limitations of deterministic models and leveraging the efficiency of approximate Bayesian inference, this research contributes to the growing field of uncertainty-aware deep learning in bioinformatics.

While previous studies have mainly focused on improving classification accuracy, they often lacked a clear mechanism for quantifying prediction uncertainty, which is essential for critical decision-making in bioinformatics. Some approaches that attempt to model uncertainty involve complex implementations or require high computational resources. In contrast, the novelty of our study lies in the integration of Monte Carlo Dropout (MCD) into a neural network architecture for DNA sequence classification, enabling both accurate predictions and scalable uncertainty estimation in a single, lightweight model. This research offers a practical and interpretable solution for uncertainty-aware classification, which has not been extensively explored in prior works.

3. METHODOLOGY

The methodology contains the technical stages that will be carried out at the research stage. Figure 1 illustrates the complete methodology pipeline of the proposed Monte Carlo Dropout (MCD)-based DNA sequence classification model. The process begins with data collection, feature-target separation, and normalization. The dataset is then split into training, validation, and testing sets.

At the core of the workflow, the figure shows the neural network architecture employed in the MCD model, comprising an input layer with 180 neurons, two hidden

layers with 128 and 64 neurons respectively (each followed by dropout layers with a dropout rate of 0.3), and an output layer with 3 neurons representing the splice junction classes (EI, IE, Neither).

This model is trained over 20 epochs using the training dataset. During the inference phase, Monte Carlo sampling with 100 forward passes is conducted while keeping dropout layers active, enabling the estimation of both mean predictions and predictive uncertainty.

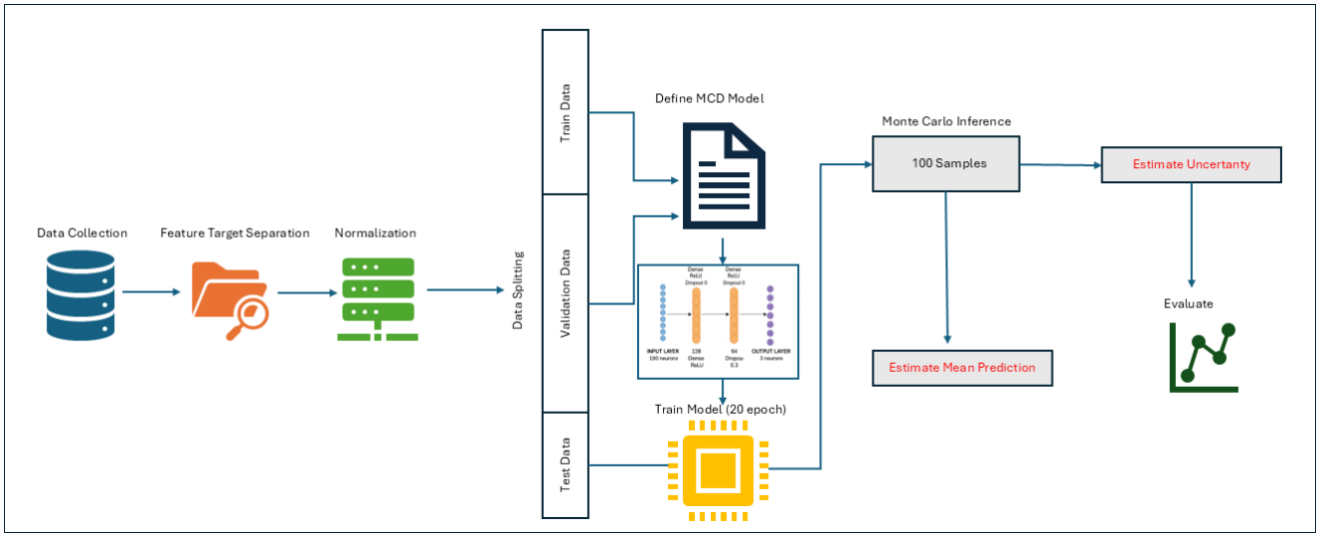

FIGURE 1. RESEARCH METHODOLOGY

These outputs are then evaluated using accuracy, F1-score, and uncertainty visualizations to validate the model's performance and interpretability Finally, the model's performance and interpretability are evaluated using classification accuracy, F1-score, and visualization of uncertainty.

3.1 Data Collection

The dataset used in this research was sourced from the Kaggle platform and contains a total of 3,187 data samples. This dataset is specifically designed for splice junction classification, a critical task in bioinformatics. Each sample represents a DNA sequence encoded using binary features, with each nucleotide (A, C, G, T) represented as a three-dimensional binary indicator variable. These features capture the unique characteristics of DNA sequences, enabling effective modelling for splice junction classification.

The dataset consists of 180 features representing the binary encoding of 60 nucleotides within each DNA sequence. In addition, there is a target column, labelled as "class," which indicates the splice junction type for each sequence. The target variable has three classes:

- **0 (EI)**: Exon-Intron boundaries (Donor sites)
- **1 (IE)**: Intron-Exon boundaries (Acceptor sites)
- **2 (Neither)**: Neither EI nor IE boundaries

Below is a summary of the features in the dataset:

TABLE 1. FEATURE DESCRIPTION

Feature Name	Description
A0, A1, A2	Binary representation of nucleotide "A"
C0, C1, C2	Binary representation of nucleotide "C"
G0, G1, G2	Binary representation of nucleotide "G"
T0, T1, T2	Binary representation of nucleotide "T"
...	... (repeated for all 60 nucleotide positions)
Class	Target class (0=EI, 1=IE, 2=Neither)

3.2 Feature Target Separation

To prepare the dataset for modeling, the features and the target variable were separated. The features (X) consist of the binary-encoded nucleotide indicators, representing the DNA sequence across 180 binary features. Each nucleotide (A, C, G, T) is encoded as a unique triplet of binary values, effectively capturing its identity in the sequence. These features are derived from 60 nucleotide positions within each DNA sequence, where every position is represented by three binary values.

The target variable (y) is extracted from the "class" column, which labels each sequence with its respective splice junction type. The target variable has three classes: (0)represents Exon-Intron (EI) boundaries (donor sites), (1)represents Intron-Exon (IE) boundaries (acceptor sites), and (2) represents sequences that do not correspond to either EI or IE boundaries (Neither).

By separating the features and the target variable, the dataset is structured for subsequent preprocessing and modeling [20]. This step ensures a clear distinction between the predictors and the response variable, enabling the model to effectively learn the relationships between the DNA sequence and its splice junction classification. The resulting feature matrix (X) contains 3187 samples with 180 features, while the target variable (y) is a single-column vector with corresponding labels for each sample.

3.3 Normalization

Normalization is an essential preprocessing step to ensure that the features in the dataset are scaled to a uniform range, improving the model's convergence and performance [21]. In this study, the features representing the binary-encoded nucleotides of the DNA sequences were normalized using the MinMaxScaler technique from the Scikit-learn library. The MinMaxScaler scales each feature to a range of 0 to 1, preserving the relationships

between the original values while ensuring that all features contribute equally to the model training process.

Given that the DNA sequence features are binary (0 or 1), normalization does not change their values but ensures compatibility with the neural network model. Neural networks are sensitive to input feature scales, and normalization helps prevent issues such as slow convergence or dominance of larger feature ranges over smaller ones.

The normalization process was applied only to the feature matrix (X), while the target variable (y) was excluded from this step as it consists of categorical labels. After normalization, the feature matrix=(X) retains its original shape of (3187×180), but all feature values are now scaled within the range of 0 to 1. This preprocessing step prepares the dataset for effective training and evaluation in subsequent stages of the methodology.

3.4 Data Splitting

To ensure an effective training and evaluation process, the dataset was divided into three subsets: training, validation, and test sets. This approach helps prevent overfitting by ensuring the model is trained on one portion of the data while being evaluated on separate, unseen subsets [22]. The splitting process was performed using the "train_test_split" function from Scikit-learn, with stratification applied based on the target variable (y) to maintain consistent class distributions across all subsets. This is particularly important in multi-class classification tasks to avoid potential class imbalance issues.

The dataset was initially split into 70% training data and 30% temporary data. The temporary data was further split equally into validation and test sets, each comprising 15% of the total data. As a result, the training set contained 2230 samples, the validation set contained 478 samples, and the test set also contained 478 samples. The training set was used to train the model, enabling it to learn patterns in the data. The validation set was employed during training to fine-tune hyperparameters and monitor model performance, while the test set was held out entirely during training and used to provide an unbiased evaluation of the final model. This structured division ensures that the training process is robust, the model's performance is well-monitored, and the final evaluation accurately reflects the model's generalization capability to unseen data.

3.5 Define And Train MCD Model

To address the task of classifying DNA splice junctions and estimating prediction uncertainty, a Monte Carlo Dropout (MCD) model was defined and trained. The MCD approach integrates dropout layers during both training and inference, enabling the model to approximate Bayesian inference and quantify prediction uncertainty. This characteristic makes MCD particularly valuable for tasks where understanding the confidence of predictions is crucial.

The architecture of the MCD model consists of a dense neural network with three key layers. The input layer is followed by a dense layer comprising 128 neurons with ReLU activation, designed to capture complex patterns within the DNA sequence features. A dropout layer with a rate of 30% introduces stochasticity, followed by another dense layer with 64 neurons and ReLU activation. Another dropout layer with the same rate is added before the output layer, which contains three neurons representing the splice junction classes (EI, IE, and Neither). The output layer uses a softmax activation function to produce probabilities for each class.

The training process optimized the following sparse categorical cross-entropy loss function $(\mathcal{L})$ for multi-class classification:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log\left(\widehat{y_{i,j}}\right) \quad (1)$$

where (N) is the number of samples, (C)is the number of classes, $(y_{i,j})$ is a binary indicator for the true class of sample (i), and $(\widehat{y_{i,j}})$ is the predicted probability for class (j) for sample (i). The Adam optimizer with a learning rate of 0.001 was used for training, and the model was trained over 20 epochs with a batch size of 32. During training, the validation set was used to monitor performance and prevent overfitting. Following training, the model's robustness was evaluated using the test set.

Inference was performed with Monte Carlo sampling, in which the dropout layers remained active. For each test sample, the model generated 100 predictions, creating a distribution of probabilities. The mean of these probabilities was used as the final class prediction, while the standard deviation provided a measure of uncertainty. This methodology enables robust classification of DNA splice junctions while simultaneously offering insights into the model's prediction confidence.

3.6 Monte Carlo Inference

Monte Carlo Inference was employed in this study to leverage the stochastic behavior of the dropout layers in the Monte Carlo Dropout (MCD) model. This method allows the estimation of prediction uncertainty by performing multiple forward passes through the model during inference, with dropout layers remaining active. This approach provides a probabilistic output for each sample, rather than a single deterministic prediction [23].

During inference, the test dataset was passed through the trained MCD model $(N = 100)$ times for each sample. For each forward pass, the dropout layers introduced randomness, producing slightly different predictions. This resulted in a distribution of predictions for each sample, which was then summarized to derive both the final prediction and its associated uncertainty. The mean of the predicted probabilities across the (N) iterations was calculated as:

$$\widehat{y_i} = \frac{1}{N}\sum_{n=1}^{N}\widehat{y_{i,n}} \quad (2)$$

where $(\widehat{y_i})$ is the final predicted probability vector for sample (i), and $(\widehat{y_{i,n}})$ is the predicted probability vector for sample (i) during the (n)-th forward pass. The uncertainty for each prediction was quantified using the standard deviation of the predicted probabilities:

$$\text{Uncertainty}_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\widehat{y_{i,n}} - \widehat{y_i}\right)^2} \quad (3)$$

The resulting mean probabilities were used to determine the final class labels for each sample by selecting the class with the highest mean probability. The standard deviation provided a quantitative measure of uncertainty, highlighting samples where the model was

less confident in its predictions. Monte Carlo Inference adds an essential interpretability layer to the classification process by allowing the identification of uncertain predictions. This capability is particularly beneficial in domains like bioinformatics, where understanding the reliability of predictions is critical for downstream analysis and decision-making. The results of the Monte Carlo Inference are presented and analyzed in the subsequent sections.

3.7 Estimate Mean Prediction

Estimating the mean prediction is a crucial step in Monte Carlo Inference, as it aggregates the multiple stochastic outputs generated by the Monte Carlo Dropout (MCD) model into a single, interpretable prediction for each sample. During this process, the model was used to perform ($N = 100$) forward passes for each test sample, with the dropout layers remaining active to introduce stochasticity in the predictions. For each sample, the predicted probability vector from each forward pass was averaged to compute the mean prediction. The formula for calculating the mean prediction for sample (i) is given as:

$$\widehat{y_i} = \frac{1}{N}\sum_{n=1}^{N} \widehat{y_{i,n}} \quad (4)$$

Where ($\widehat{y_i}$) is the mean predicted probability vector for sample(i), (N) is the total number of Monte Carlo samples (100 in this case) and ($\widehat{y_{i,n}}$) is the predicted probability vector for sample (i) from the (n)-th forward pass. The mean prediction provides the final probability distribution over the three classes: Exon-Intron (EI), Intron-Exon (IE), and neither. The class with the highest probability in the mean prediction vector is selected as the final predicted class for the sample:

$$\text{Class}_i = \text{argmax}(\widehat{y_i}) \quad (5)$$

This aggregation ensures that the stochastic nature of the dropout layers during inference contributes to the robustness of the predictions. By averaging the outputs, the model's prediction becomes less sensitive to individual stochastic variations and better reflects the overall confidence of the model. The estimated mean predictions form the basis for evaluating the classification accuracy of the MCD model. They also serve as a foundation for calculating the uncertainty associated with each prediction, which is discussed in the next section.

3.8 Estimate Uncertainty

Estimating uncertainty is a vital aspect of Monte Carlo Dropout (MCD), providing insights into the confidence of the model's predictions. This step quantifies the variability in predictions generated during Monte Carlo Inference by calculating the standard deviation of the predicted probabilities across multiple forward passes. By doing so, it enables the identification of predictions where the model is less confident, which is critical for tasks requiring high reliability and interpretability. For each test sample, the uncertainty was computed as the standard deviation of the predicted probabilities across ($N = 100$) forward passes. Mathematically, the uncertainty for sample (i) is calculated as:

$$\text{Uncertainty}_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\widehat{y_{i,n}} - \widehat{y_i}\right)^2} \quad (6)$$

Where ($\widehat{y_{i,n}}$) is the predicted probability vector for sample (i) from the \(n\)-th forward pass, ($\widehat{y_i}$)is the mean predicted probability vector for sample (i) (as calculated in Section 3.7) and (N) is the total number of Monte Carlo samples (100 in this study). The resulting uncertainty values provide a numerical measure of the model's confidence in its predictions. Higher uncertainty indicates that the model's predictions vary significantly across different forward passes, suggesting that the sample might be ambiguous or challenging for the model to classify. Conversely, lower uncertainty implies that the model is consistently confident in its prediction.

The estimated uncertainty values were used to identify samples with high prediction variability, which can inform further analysis or data augmentation efforts. For example, samples with high uncertainty could indicate areas where additional data collection or refinement of the model might be beneficial. This capability adds an essential interpretability layer to the classification process, making the MCD approach particularly valuable for applications in bioinformatics and other domains requiring robust decision-making.

3.9 Evaluate

The evaluation phase is critical to assess the performance of the Monte Carlo Dropout (MCD) model in classifying DNA splice junctions. This step involves measuring the model's predictive accuracy and analyzing its ability to quantify uncertainty effectively. The evaluation process uses the test set, which was held out entirely during training and validation, to ensure an unbiased assessment of the model's generalization capability. The classification accuracy was computed by comparing the predicted class labels ($\text{argmax}(\widehat{y_i})$) derived from the mean predictions with the true class labels in the test set. Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \quad (7)$$

The evaluation also included an analysis of uncertainty for each prediction. The uncertainty values, computed as the standard deviation of predictions across 100 Monte Carlo samples, were analyzed to identify samples where the model was less confident. High-uncertainty samples were further investigated to understand their characteristics and their potential impact on the model's performance. Furthermore, the confusion matrix was used to provide a detailed breakdown of the model's classification results across the three classes: Exon-Intron (EI), Intron-Exon (IE), and neither. This analysis highlighted the model's strengths and weaknesses in distinguishing between the classes.

The combined evaluation of accuracy, precision, recall, F1-score, and uncertainty provide a comprehensive understanding of the model's predictive performance and reliability. This evaluation framework ensures that the MCD model is not only accurate but also interpretable and robust, addressing the critical requirements of DNA sequence classification tasks. In addition to numerical evaluation metrics, the classification results are illustrated through a confusion matrix, as shown in Figure 4. The matrix presents a clear overview of the model's performance across the three classes. Most predictions lie on the diagonal, indicating that the model correctly

classifies most samples. The low values in the off-diagonal cells suggest minimal misclassification, demonstrating the model's capability to distinguish between Exon-Intron, Intron-Exon, and Neither splice junctions effectively.

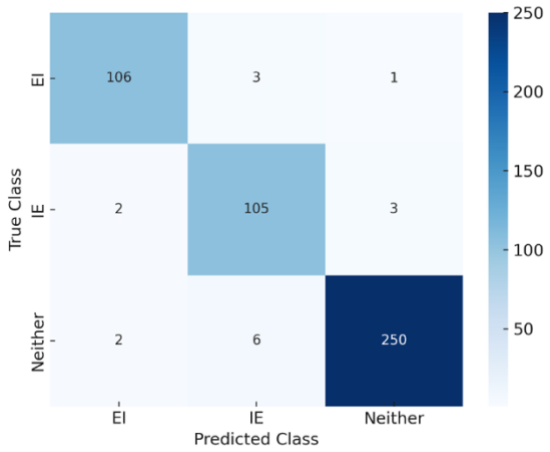

FIGURE 2. CONFUSION MATRIX OF THE MCD MODEL

4. RESULT AND DISCUSSION

The results of this study demonstrate the performance of the baseline neural network (NN) and the Monte Carlo Dropout (MCD) model for classifying DNA splice junctions. The comparison highlights both the classification accuracy and the ability of the MCD model to estimate prediction uncertainty. The baseline neural network achieved a test loss of 0.1229 and a test accuracy of 95.61%. These results indicate that the model is highly capable of learning and generalizing the patterns within the DNA dataset. However, the baseline NN provides deterministic predictions and does not account for uncertainty in its outputs, which is a limitation in bioinformatics applications requiring confidence estimation.

The Monte Carlo Dropout model achieved a slightly higher accuracy of 96.03%, demonstrating improved performance over the baseline NN. Additionally, the MCD model provides uncertainty estimates for each prediction. For example, the uncertainty values for the first five test samples are Table 2:

TABLE 2. UNCERTAINTY VALUES SAMPLE

Sample	Uncertainty		
	Class 0	Class 1	Class 2
1	34	1097	1097
2	4384	0,000577	4384
3	1818	2317	3569
4	1177	2345	117
5	0,003601	3893	3922

These values highlight the variability in prediction confidence for different samples. Lower uncertainty values indicate high confidence, while higher values suggest uncertainty in the model's prediction. The uncertainty distribution for all test samples is shown in Figure 3. The histogram reveals that most predictions have very low uncertainty, with most uncertainty values clustering near zero. This suggests that the MCD model is confident in its predictions for a significant portion of the dataset. However, there are a few samples with higher uncertainty, indicating cases where the model was less confident in its predictions.

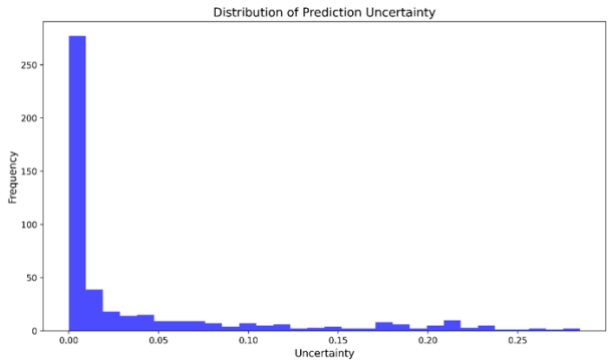

FIGURE 3. DISTRIBUTION OF PREDICTION UNCERTAINTY

The presence of high-uncertainty samples could be attributed to ambiguous patterns in the DNA sequences or potential overlap between the splice junction classes (EI, IE, Neither). Such samples warrant further investigation, as they could highlight areas where additional data or model refinement may be beneficial. The results of the study show that the Monte Carlo Dropout (MCD) model slightly outperformed the baseline neural network (NN) in terms of classification accuracy, achieving 96.03% compared to 95.61%. This indicates that the MCD model not only matches the performance of a standard NN but also provides additional insights through uncertainty estimation. Unlike the baseline NN, which generates deterministic predictions, the MCD model quantifies the confidence of its predictions by estimating uncertainty. This feature enhances the interpretability and reliability of the classification results, particularly in applications where understanding prediction confidence is crucial.

The MCD model's performance was further evaluated using precision, recall, and F1-score for each class, as summarized in the following results:

TABLE 3. EVALUATION

Metric	Class 0 (EI)	Class 1 (IE)	Class 2 (Neither)	Macro Avg	Weighted Avg
Precision	0.94	0.94	0.98	0.95	0.96
Recall	0.97	0.96	0.96	0.96	0.96
F1-Score	0.96	0.95	0.97	0.96	0.96
Support	115	115	248	-	-

The overall accuracy of the model was 96%, demonstrating its strong performance in classifying DNA splice junctions. The high precision across all classes indicates that the model minimizes false positives, while the high recall ensures that it correctly identifies most of the positive samples. The F1-score, which balances precision and recall, consistently exceeds 0.95 for all classes, reflecting the model's robustness.

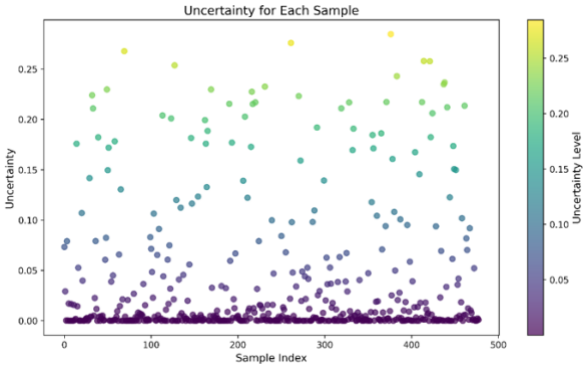

FIGURE 4. UNCERTAINTY FOR EACH SAMPLE

Figure 4 provides a detailed visualization of the prediction uncertainty for each test sample. The scatter plot displays the uncertainty values on the y-axis, with each point representing a single test sample. The color intensity indicates the level of uncertainty, with lighter colors representing higher uncertainty. Most samples exhibit low uncertainty values, clustered near 0, indicating the model's high confidence in its predictions. However, a few samples have significantly higher uncertainty, as evidenced by the scattered points at higher y-axis values. These high-uncertainty samples might correspond to ambiguous or challenging cases, where the distinction between classes is less clear. Identifying these samples provides valuable insights for improving the model, such as through data augmentation or refinement of the model architecture.

The results demonstrate the effectiveness of Monte Carlo Dropout (MCD) in classifying DNA splice junctions while providing uncertainty estimates. The MCD model achieved an accuracy of 96.03%, slightly outperforming the baseline neural network (NN) at 95.61%. This highlights the ability of MCD to maintain high predictive accuracy while quantifying uncertainty, a critical feature for tasks requiring interpretability and confidence estimation. The uncertainty analysis, visualized in Figure 3 and Figure 4, reveals that most predictions exhibit low uncertainty, indicating high confidence in the model's classifications. However, a subset of samples with higher uncertainty suggests ambiguous patterns or overlapping class boundaries, providing opportunities for refinement through targeted data augmentation or improved feature extraction. Evaluation metrics, including precision, recall, and F1-scores, further affirm the robustness of the MCD model across all classes. The balanced performance across the dataset demonstrates the model's reliability in handling varying class distributions.

Furthermore, a closer analysis was conducted on the samples with the highest uncertainty scores and misclassified outputs. We observed that most of these samples belonged to either the Exon-Intron (EI) or Intron-Exon (IE) classes, with misclassifications often occurring between these two. This may be attributed to the biological similarity between donor and acceptor sites, which can share overlapping nucleotide patterns, leading to model confusion. Additionally, these sequences frequently exhibited less distinctive binary encoding across critical positions, suggesting weaker signals in their input features. The high-uncertainty samples also tended to fall near the decision boundary in the model's output probability space, with no dominant class prediction. These findings highlight the importance of enhancing training data quality, possibly by including extended sequence context or domain-specific sequence motifs, to improve model confidence and reduce ambiguities in future iterations.

5. CONCLUSIONS

This research demonstrates that Monte Carlo Dropout (MCD) is an effective approach for classifying DNA splice junctions while providing interpretable uncertainty estimates. By incorporating dropout layers during inference, the MCD model achieves high accuracy and enables the quantification of prediction confidence, addressing the critical need for interpretability in bioinformatics tasks. The ability to identify uncertain predictions offers a practical tool for improving model reliability and guiding future efforts, such as refining data quality or enhancing feature engineering. Moving forward, this research can be extended by exploring alternative Bayesian inference methods or integrating domain-specific biological knowledge to further improve model performance. Additionally, applying this approach to other genomic tasks, such as gene expression prediction or variant classification, can unlock new possibilities for advancing bioinformatics and computational biology.

REFERENCES

[1] M. Sha and M. P. Rahamathulla, "Splice site recognition-deciphering Exon-Intron transitions for genetic insights using Enhanced integrated Block-Level gated LSTM model," *Gene*, vol. 915, p. 148429, 2024.

[2] A. J. Clark and J. W. Lillard Jr, "A comprehensive review of bioinformatics tools for genomic biomarker discovery driving precision oncology," *Genes*, vol. 15, no. 8, p. 1036, 2024.

[3] A. Subasi, "DNA sequence classification using artificial intelligence," in *Applications of Artificial Intelligence Healthcare and Biomedicine*, Elsevier, 2024, pp. 401–415.

[4] M. M. Uddin, J. Shiddike, A. Ahmed, and T. Ahsan, "Promoter Prediction in DNA Classification Using Machine Learning Algorithms," in *2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, IEEE, 2024, pp. 254–260.

[5] K. Chandrashekar, V. Niranjan, A. Vishal, and A. S. Setlur, "Integration of artificial intelligence, machine learning and deep learning techniques in genomics: review on computational perspectives for NGS analysis of DNA and RNA seq data," *Curr. Bioinforma.*, vol. 19, no. 9, pp. 825–844, 2024.

[6] Y. Ye *et al.*, "Machine learning-based classification of deubiquitinase USP26 and its cell proliferation inhibition through stabilizing KLF6 in cervical cancer," *Comput. Biol. Med.*, vol. 168, p. 107745, 2024.

[7] V. Nerkar, V. Kimbahune, and R. Somkunwar, "DNA Sequence Binding to specified Protein Strutcure and Classification by using CNN and k-mer Encoding Model," in *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, IEEE, 2024, pp. 1–6.

[8] N. Y. Ahmed *et al.*, "An Efficient Deep Learning Approach for DNA-Binding Proteins Classification from Primary Sequences," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, pp. 1–14, 2024.

[9] K.-H. Chao, A. Mao, S. L. Salzberg, and M. Pertea, "Splam: a deep-learning-based splice site predictor that improves spliced alignments," *Genome Biol.*, vol. 25, no. 1, p. 243, 2024.

[10] M. Ali, D. Shah, S. Qazi, I. A. Khan, M. Abrar, and S. Zahir, "An effective deep learning-based approach for splice site identification in gene expression," *Sci. Prog.*, vol. 107, no. 3, p. 00368504241266588, 2024.

[11] L. Rentao, L. Yelin, G. Lixin, and L. Mengshan, "Predicting DNA sequence splice site based on graph convolutional network and DNA graph construction," *J. King Saud Univ.-Comput. Inf. Sci.*, p. 102089, 2024.
[12] L. Ngartera, M. A. Issaka, and S. Nadarajah, "Application of Bayesian Neural Networks in Healthcare: Three Case Studies," *Mach. Learn. Knowl. Extr.*, vol. 6, no. 4, pp. 2639–2658, 2024.
[13] T. Yu, Z. Zou, and H. Xiong, "Can Uncertainty Quantification Enable Better Learning-based Index Tuning?," *ArXiv Prepr. ArXiv241017748*, 2024.
[14] R. Y. Oh *et al.*, "A systematic assessment of the impact of rare canonical splice site variants on splicing using functional and in silico methods," *Hum. Genet. Genomics Adv.*, vol. 5, no. 3, 2024.
[15] C. Firtina *et al.*, "Aphmm: Accelerating profile hidden markov models for fast and energy-efficient genome analysis," *ACM Trans. Archit. Code Optim.*, vol. 21, no. 1, pp. 1–29, 2024.
[16] A. O. Abhadionmhen *et al.*, "Machine Learning Approaches for Microorganism Identification, Virulence Assessment, and Antimicrobial Susceptibility Evaluation Using DNA Sequencing Methods: A Systematic Review," *Mol. Biotechnol.*, pp. 1–29, 2024.
[17] A. P. Avila Santos *et al.*, "BioDeepfuse: a hybrid deep learning approach with integrated feature extraction techniques for enhanced non-coding RNA classification," *RNA Biol.*, vol. 21, no. 1, pp. 1–12, 2024.
[18] T. Gudavarthi, S. V. R. Seri, A. C. Thadoju, P. K. Sarangi, and M. Jabbar, "DNA Sequence Classification Using RRCNN and LSTM," in *2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, IEEE, 2024, pp. 1–6.
[19] G. He, Y. Zhao, and C. Yan, "Uncertainty quantification in multiaxial fatigue life prediction using Bayesian neural networks," *Eng. Fract. Mech.*, vol. 298, p. 109961, 2024.
[20] A. G. Putrada, I. D. Oktaviani, M. N. Fauzan, and N. Alamsyah, "CNN Pruning for Edge Computing-Based Corn Disease Detection with a Novel NG-Mean Accuracy Loss Optimization," *Telematika*, vol. 17, no. 2, pp. 68–83, 2024.
[21] N. Alamsyah, A. P. Kurniati, and others, "A Novel Airfare Dataset To Predict Travel Agent Profits Based On Dynamic Pricing," in *2023 11th International Conference on Information and Communication Technology (ICoICT)*, IEEE, 2023, pp. 575–581.
[22] N. Alamsyah, T. P. Yoga, B. Budiman, and others, "IMPROVING TRAFFIC DENSITY PREDICTION USING LSTM WITH PARAMETRIC ReLU (PReLU) ACTIVATION," *JITK J. Ilmu Pengetah. Dan Teknol. Komput.*, vol. 9, no. 2, pp. 154–160, 2024.
[23] N. Alamsyah, B. Budiman, T. P. Yoga, and R. Y. R. Alamsyah, "XGBOOST HYPERPARAMETER OPTIMIZATION USING RANDOMIZEDSEARCHCV FOR ACCURATE FOREST FIRE DROUGHT CONDITION PREDICTION," *J. Pilar Nusa Mandiri*, vol. 20, no. 2, pp. 103–110, 2024.

AUTHORS

Nur Alamsyah
A faculty member in the Information Systems Program at the Faculty of Technology and Informatics, Universitas Informatika dan Bisnis Indonesia. His research focuses on Data Science, Mathematical Modeling, and Machine Learning. He is particularly interested in applying these fields to areas such as business analytics, predictive modeling, and computational simulations. His work aims to drive innovation in data-driven strategies and machine learning applications, fostering advancements in both academic exploration and practical problem-solving to support informed decision-making and optimize operational processes.

Budiman
This researcher has expertise in machine learning and data science. Currently affiliated with Universitas Informatika dan Bisnis Indonesia, Bandung. This researcher has a strong academic background in informatics and experience in designing data-driven solutions. His study focuses on applying algorithms for big data analysis to support better business decision-making. In addition, he is proficient in using several tools to process and analyse data effectively. This researcher continues to develop skills and knowledge in data-driven technology and innovation. He is committed to making significant contributions to computer science and technology.

Reni Nursyanti
is a lecturer at the Informatics Study Programme, Universitas Informatika dan Bisnis Indonesia (UNIBI). With a strong academic background in information technology, I have expertise in software development, data analysis, and technology innovation. As a dedicated educator I always encourage students to think critically and creatively. Besides teaching, I am also active in research and community service, especially in the application of technology to help MSMEs and the education sector. My interactive teaching approach has earned me respect among students and peers.

Elia Setiana

Faculty member in Informatics Program at Faculty of Technology and Informatics, Universitas Informatics and Business Indonesia. His research focuses on Data Science, software engineering, and system security. She is particularly interested in applying these fields to areas such as business analysis and software projects. Her work aims to drive innovation in application strategies and advancements in academic exploration and problem solving.

Venia Restreva Danestiara

A dedicated lecturer in the field of Informatics at Universitas Informatika dan Bisnis Indonesia, with expertise in Data Science and Molecular Docking, specializing in the development of machine learning-based scoring functions for predicting antiviral inhibitors of specific diseases. Her research also extends to deep learning applications in image data processing, leveraging artificial intelligence to address complex challenges in healthcare and bioinformatics.