Exploring the Intersection of Information Theory and Machine Learning

Yousef Jaradat
Department of Electrical Engineering
Al-Zaytoonah University of Jordan
y.jaradat@zuj.edu.jo

Mohammad Masoud
Department of Electrical Engineering
Al-Zaytoonah University of Jordan
m.zakaria@zuj.edu.jo

Ahmad Manasrah
Department of Mechanical Engineering
Al-Zaytoonah University of Jordan
ahmad.mansrah@zuj.edu.jo

Mohammad Alia
Department of Cybersecurity
Al-Zaytoonah University of Jordan
dr.m.alia@zuj.edu.jo

Khalid Suwais
Faculty of Computer Studies
Arab Open University, Saudi Arabia
Khaled.suwais@arabou.edu.sa

Sally Almanasra
Faculty of Computer Studies
Arab Open University, Saudi Arabia
s.almanasra@arabou.edu.sa

Abstract: *This study addresses the need for a unified framework demonstrating Information Theory's (IT) pervasive impact across diverse Machine Learning (ML) tasks. We investigate how IT principles-including entropy, Mutual Information (MI), cross-entropy, KL-divergence, and Information Gain (IG)-rigorously guide ML model design, optimization, and interpretability. Our approach combines theoretical elucidation with empirical validation on standard benchmarks. IT enhances feature selection; for instance, MI-ranked features in the breast cancer dataset improved classifier accuracy to 95.1% (top 20) and 93% (top 5), outperforming F-score selection. It also improves model training; cross-entropy loss in Neural Networks (NNs) for Iris classification led to faster convergence and high accuracy (0.98 training, 0.95 validation), surpassing MSE loss. For generative models, KL-divergence effectively structures Variational Auto-Encoder (VAE) latent spaces from Modified National Institute of Standards and Technology (MNIST) data, promoting compact, continuous representations ideal for generation. Finally, the Information Bottleneck (IB) principle, applied to Canadian Institute For Advanced Research (CIFAR-100), yielded competitive test accuracy (51% vs. 50% for baseline Convolutional Neural Network. (CNN)) and reduced training time (925.02s vs. 1015.75s), highlighting its efficacy in learning compressed, predictive representations. These findings collectively underscore its continued crucial role as a unifying paradigm for addressing fundamental challenges in the evolving ML ecosystem, providing solutions for feature selection, model robustness, and generalization.*

Keywords: *Information theory, machine learning, entropy, mutual information, information bottleneck.*

Received February 17, 2025; accepted June 22, 2025
https://doi.org/10.34028/iajit/22/5/1

1. Introduction

Information Theory (IT), originating from Claude Shannon's pivotal work in the 1940s, provides a fundamental framework for understanding and quantifying information and uncertainty. It offers a robust set of tools, such as entropy and Mutual Information (MI), which are not merely theoretical constructs but have practical applications spanning diverse fields from communication system optimization to data security [8]. In the realm of Machine Learning (ML), IT is incredibly important. It equips us with essential metrics to evaluate and fine-tune algorithms for enhanced efficiency and effectiveness. For instance, entropy quantifies data uncertainty, aiding in the selection of optimal features for model input. MI reveals the statistical dependence between variables, proving invaluable for tasks like feature selection and dimensionality reduction. Derived from entropy, Information Gain (IG) is critical in Decision Trees (DTs) for determining the most effective data splits, leading to more accurate and interpretable models [3].

ML algorithms frequently encounter challenges such as noisy data, optimal feature selection, and model optimization. IT directly addresses these issues; for example, cross-entropy is widely used as a loss function in classification tasks to measure the alignment between predicted and true outcomes, guiding the training of Neural Networks (NNs). Similarly, MI is leveraged in clustering algorithms to assess the quality of data groupings, ensuring meaningful clusters [20, 23].

While IT utility in specific ML tasks is well-established, a comprehensive analysis of its unifying principles across diverse modern architectures and datasets, particularly in addressing challenges like model interpretability and robustness to noisy data, remains an area requiring systematic investigation. Despite the proliferation of increasingly complex generative models, the underlying relevance and advanced applications of IT are not always thoroughly explored or demonstrated across various modern ML paradigms. This paper addresses this gap by providing a unified framework that systematically demonstrates the pervasive impact of IT concepts across diverse ML tasks, moving beyond isolated applications. We aim to rigorously investigate

how foundational IT principles guide the design, optimization, and interpretability of ML models showcasing their continued relevance and power in the contemporary data science landscape, [11]. The rest of this paper is organized as follows: Section 2 presents mathematically the fundamentals of IT. Section 3 provides an overview of ML concepts and algorithms. Section 4 discusses the applications of IT in ML, supported by empirical results on benchmark datasets. Finally, section 5 offers a conclusion, summarizing the findings and outlining future research directions.

2. Theoretical Background

This section provides the necessary theoretical background of all IT concepts applied in ML models.

2.1. Shannon Information Content

Shannon information content, $I(x)$, also known as self-information, describes the quantity of information acquired by observing an event. The information content, $I(x)$, can be defined using the following Equation (1).

$$I(x) = -logP(x) \quad (1)$$

$P(x)$ represents the probability value of the event x. For instance, in the scenario when a coin consistently displays heads, the outcomes are expected, and no new information is gained. Thus, the amount of information is zero. If the coin is fair, the probability distribution is evenly distributed and it is extremely unlikely whether the next outcome will be heads or tails. Thus, the level of information content is at its highest. In computer science, information content is defined as the minimum number of bits necessary to encode information efficiently. We utilize base 2 for the logarithm function. Hence, the information content for a fair coin flip is calculated as $-log_2(½)$, which equals 1. That is to say, we acquire 1-bit of information with each individual coin flip.

A variable X, is considered a random variable if it represents a value that comes from a random process, like the number that appears when you roll a die or the total number of heads you get after flipping a coin a number of times. We can describe the value of this random variable, X, using a probability distribution, $p(X)$. For instance, if X represents the total sum of rolling five dice, we can use a Gaussian distribution to model X, thanks to the central limit theorem [1].

2.2. Shannon Entropy

Entropy, $H(X)$, which was proposed in 1948 by Claude Shannon, quantifies uncertainty associated with a Discrete Random Variable (DRV), X, by assigning a scalar value to it [19]. A DRV, X, with Probability Mass Function (PMF) $P(x)$ has an entropy $H(X)$ given by:

$$H(X) = -\sum_{x \in X} P(x) logP(x) \quad (2)$$

Where:

- X represents the set of all possible values that the DRV, X, can take.
- x is a specific value within the set of possible outcomes X.
- $P(x)$ is the *pmf* of the variable X with all values x.
- The summation ($\sum_{x \in X}$) signifies that the operation is performed by summing the set of all values of x.

If the alogarithm is base 2, the entropy is measured in bits. For example, in flipping a fair coin, $P(X=H)=½$ and $P(X=T)=½$. Its entropy equals $H(X)=-(P(X=H))log+P(X=T)log(P(X=T))=1$. The mean value of the content is one bit. In other words, we require an average of one bit to represent or encode a single event. Entropy sets a minimum limit for the average number of bits required to encode events based on the probability distribution P. Figure 1 shows the graph of the entropy as a function of P to understand how entropy changes with the probability of the outcomes for a binary random variable X.

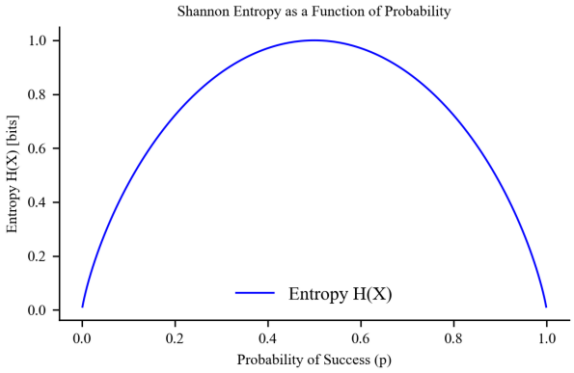


Figure 1. Entropy change of two states random variable.

The Shannon curve in Figure 1 shows the following points:

- Maximum entropy: the curve attains its highest value when $p=0.5$. This is the point at which the level of uncertainty is the maximum, as both outcomes (0 and 1) are equally probable. The binary variable has a maximum entropy value of 1 bit.
- Zero entropy: the curve intersects the zero point at p=0 and p=1. One outcome is certain at these instances, eliminating any uncertainty. When the value of p is 0, the outcome is consistently 1, and when the value of p is 1, the outcome is consistently 0.
- Curve symmetry: the entropy function exhibits symmetry with respect to p=0.5. This symmetry demonstrates that the entropy of a binary variable with probability p, denoted as $H(p)$, is equal to the entropy with probability $1-p$, denoted as $H(1-p)$.

Figure 1 visually demonstrates that entropy quantifies uncertainty, peaking when probabilities are evenly distributed and diminishing as one outcome becomes more predictable.

2.3. Joint and Conditional Entropy

Joint entropy, $H(X, Y)$, quantifies the total uncertainty associated with two random variables X and Y. It's calculated using their joint probability distribution:

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} P(x,y)logP(x,y) \quad (3)$$

Joint entropy quantifies the combined level of uncertainty when evaluating two variables simultaneously. Consider the scenario of simultaneously flipping two coins. Each coin has two possible outcomes: heads or tails. Therefore, when two coins are flipped, there are four possible outcomes: (heads, heads), (heads, tails), (tails, heads), and (tails, tails). Joint entropy quantifies the amount of information required to describe the outcomes of two-coin flips collectively.

Conditional entropy, $H(Y/X)$, measures the uncertainty in Y given that X is known. The formula is:

$$H(Y \mid X) = -\sum_{x\in X} P(x) \sum_{y\in Y} P(y \mid x)logP(y \mid x) \quad (4)$$

Conditional entropy measures the remaining uncertainty about a variable, given the knowledge of another variable. Consider a scenario where you have a box with balls of various colors, and you know that the box itself is of a blue color. Conditional entropy quantifies the remaining level of uncertainty regarding the color of the balls inside the box, even after having knowledge of the box color.

2.4. Mutual Information

Mutual information, $I(X; Y)$, quantifies the extent to which knowledge of one variable decreases uncertainty about another. Put simply, it measures the quantity of information that one random variable possesses regarding another random variable. When two variables are independent, their MI is equal to zero. When they rely completely on each other, the level of MI is considerable. MI is defined as:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \quad (5)$$

or using the conditional entropy:

$$I(X;Y) = H(Y) - H(Y \mid X) \quad (6)$$

Consider a scenario where you toss two coins, labeled X and Y. If the coin flips are entirely stochastic and independent, the knowledge of whether X is heads or tails provides no information about Y. In this scenario, the MI between X and Y is equal to zero. However, if there is a rule that states that Y and X always match, then the MI would be larger because knowing X enables you know Y.

2.5. Cross-Entropy

Cross-entropy quantifies the deviation between two probability distributions for a given set of events. It is frequently employed in the context of ML to assess the degree to which the predicted probabilities correspond to the actual probabilities. Better predictions are indicative of lower cross-entropy values, as the predicted distribution is more closely aligned with the true distribution.

Cross-entropy quantifies the divergence between two probability distributions, P (true distribution) and Q (predicted distribution). Cross-entropy is given by the following Equation (7):

$$H(P,Q) = -\sum_{x} P(x)logQ(x) \quad (7)$$

In simple terms, it adds up the differences between what your model predicts and what actually happens.

2.6. KL-Divergence

KL-divergence tells us how much information is lost when using one probability distribution (your predictions) to approximate another (the actual outcomes). The formula is:

$$D_{KL}(P \mid\mid Q) = \sum_{x} P(x)\log[\frac{Q(x)}{P(x)}] \quad (8)$$

Imagine you have two sets of probabilities about whether it will rain. One set represents what actually happens (e.g., 70% chance of rain, 30% chance of no rain), and the other set is your prediction (e.g., 60% chance of rain, 40% chance of no rain). KL-divergence measures how different your predicted probabilities are from the actual probabilities. If the KL-divergence is high, it means your predictions were pretty far off.

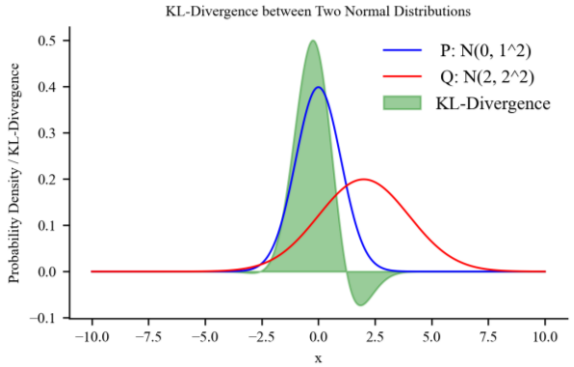


Figure 2. KL-divergence between two normal distributions.

Figure 2 illustrates the KL-divergence between two normal distributions: A "true distribution P" and a "predicted distribution Q". The shaded green area visually emphasizes the discrepancy between these two distributions, representing the magnitude of the KL-divergence. The KL-divergence is elevated in regions where Q either significantly overestimates or underestimates distribution P, underscoring the areas where Q inadequately approximates the true distribution P. A larger shaded area indicates a greater divergence, signifying a larger information loss when using Q to approximate P. This visual representation helps in understanding how KL-divergence quantifies the

dissimilarity between probability distributions, a crucial concept in generative models like Variational Auto-Encoders (VAEs) where the goal is to minimize this divergence between learned and prior distributions.

2.7. Information Gain

IG involves determining which specific piece of knowledge is most valuable in making optimal decisions. Consider determining whether it is advisable to engage in outdoor activities today. Various indicators, such as weather conditions, temperature, and humidity, are available as clues. IG reveals which clue is the most beneficial. For instance, if having knowledge of the weather conditions, such as whether it is sunny or rainy, enables you to make more informed decisions compared to knowing just the temperature, then the weather is considered to be more informative and possesses a greater IG.

Information gain $IG(Y,X)$ is calculated as:

$$IG(Y,X) = H(Y) - H(Y \mid X) \tag{9}$$

where $H(Y/X)$ is the conditional entropy of Y given the feature X and $H(Y)$ is the entropy of the target variable Y. Figure 3 shows the dependency flowchart of different types of IT measures on each other. The flowchart shows the interconnectedness of different IT metrics in a hierarchical structure. Beginning with the fundamental idea of probability distributions, it branches out into information content (self-information). An essential measure of uncertainty, entropy, is derived from this. On top of entropy exist other measurements like conditional entropy and joint entropy. cross-entropy and KL-divergence depend on probability distributions in addition to Entropy, whereas MI integrates ideas from conditional entropy and entropy. in the end, one of the most important ways that entropy and conditional Entropy are used while making decisions is through IG The above structure helps in comprehending the relationship and dependencies among these key concepts, making it easier to see how each measure builds on more fundamental ideas.

3. Machine Learning Overview

ML is an Artificial Intelligence (AI) branch where the system learns through historical data [5, 16]. Rather than being explicitly programmed for every task, ML algorithms are designed to identify patterns and relationships within data, allowing them to improve their performance and make predictions or decisions on unseen information. This iterative process of learning from experience enables models to become increasingly intelligent and accurate. In this way, the model automatically learns and improves its decision and predictive ability [10]. Deep Learning (DL) is a prominent subfield of ML that employs multi-layered artificial NNs as its computational backbone. These networks are engineered to process data through numerous interconnected layers, effectively learning hierarchical representations. Deep NNs are trained to accept representations to classify data, perform predictions, or generate outputs. The rapid proliferation of DL has been fostered by several factors, including the exponential growth of available data, the decreasing cost of parallel computing, and advancements in statistical reasoning. NNs have gained significant recognition as powerful feature extractors in advanced domains such as natural language processing, speech processing, visual object recognition, and search engine results [9, 18]. Figure 4 visually represents the interconnectedness and hierarchical relationships among AI, ML, and DL using a Venn diagram. AI is depicted as the broadest field, encompassing the development of intelligent machines capable of simulating human-like thought and behavior. ML is shown as a significant subset of AI, specifically focusing on algorithms that enable systems to learn from data without explicit programming. DL, in turn, is presented as a subfield of ML, distinguished by its use of multi-layered NN to achieve advanced learning capabilities. The concentric nature of these circles highlights that all DL models are also ML models, and all ML models fall under the umbrella of A. This illustration clarifies the scope and relationships of these key concepts within the broader field of AI.

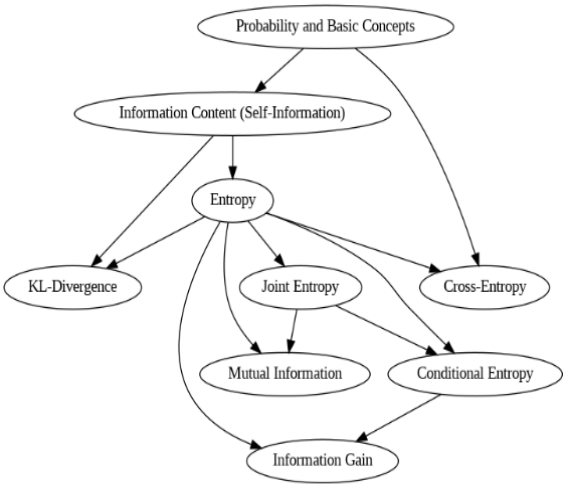


Figure 3. Information theory measures dependency flowchart.

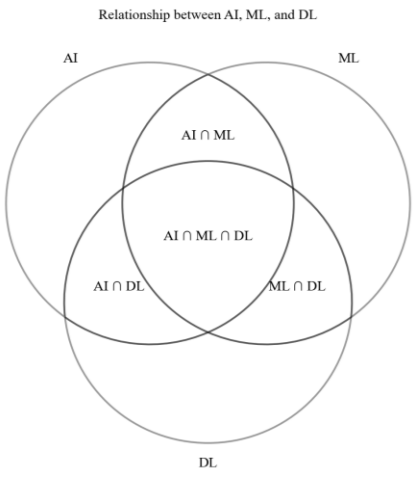


Figure 4. AI, ML, and DL relationships.

ML can be classified into the following categories, [2, 21].

- Supervised learning: this technique involves training algorithms with a labeled dataset, wherein each training example is linked to an output label. The goal is to acquire a correspondence from inputs to outputs [15]. It includes two sub-categories: Classification which refers to the prediction of a categorical label. As an example, consider email spam detection, and regression which refers to the prediction of a continuous value. For example, predicting home prices.
- Unsupervised learning: in this technique algorithms are trained on data with no labels. The goal is to deduce the dataset's natural structure [14]. It contains two sub-categories: Clustering which is the process of organizing data points into clusters. As an example, consider customer segmentation, and dimensionality reduction, which is the process of reducing the number of features while retaining critical information. As an example, consider Principal Component Analysis (PCA) [12].
- Semi-supervised learning: in this technique algorithms are trained on a combination of labeled and unlabeled data. This is beneficial when labeling data is costly or time-consuming. As an example, consider a text classification model that improves performance by combining a few classified and numerous unlabeled examples.
- Self-supervised learning: a type of unsupervised learning in which data provides supervision. It means making predictions about some data based on other data. For example, during language model training, one can predict the next word in a sentence.
- Reinforcement learning: in this strategy, an agent learns to make decisions by taking actions in an environment that maximizes some kind of cumulative reward. For example, a robot can learn to navigate a maze by receiving rewards for reaching the finish line and penalties for hitting walls.

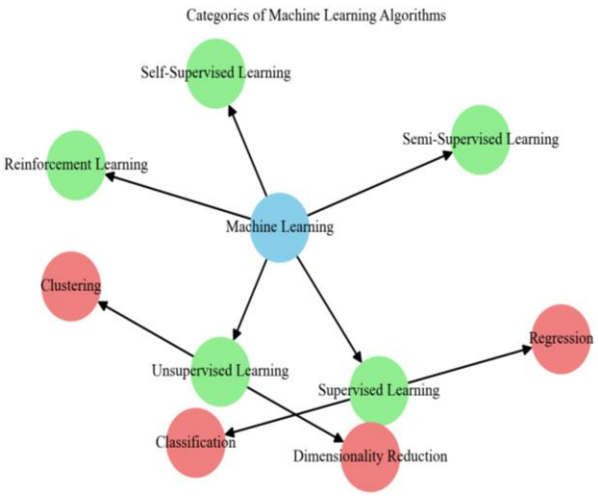


Figure 5. ML categories.

Figure 5 provides a conceptual diagram illustrating the major categories of ML algorithms. Centered around ML the diagram branches out to show its primary paradigms: Supervised learning, unsupervised learning, semi-supervised learning, self-supervised learning, and reinforcement learning. Each of these main categories then branch further into their common sub-categories, such as classification and regression under supervised learning, and clustering and dimensionality reduction under unsupervised learning. The arrows indicate relationships and classifications, offering a clear visual overview of the different approaches to ML tasks and how they relate to the central concept.

In the rapidly evolving landscape of ML and DL, several key concerns and current trends warrant attention. These include the increasing computational demands of training large models, the crucial need for enhanced model interpretability to understand decision-making processes, and significant challenges related to data privacy and security. IT, as explored in this paper, provides powerful tools and theoretical foundations that can directly contribute to addressing these contemporary issues, enabling the development of more efficient, transparent, and secure ML systems [13].

4. Applications of Information Theory in Machine Learning

IT principles play a crucial role in numerous areas of ML and DL. Below is a comprehensive explanation on the utilization of these methods, accompanied by example cases employing genuine datasets.

4.1. Entropy and Information Gain in Tree-Based Models

Tree-based models such as, DTs, Random Forests (RFs) an ensemble of DT and boosted trees like XGBoost, are a non-parametric supervised learning method used for classification and regression [6]. They function by recursively partitioning the dataset based on feature values to create a tree-like structure where each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label (or a predicted value in regression). The core principle behind constructing an effective DT and RF is to select the features that best split the data, leading to the most homogeneous child nodes with respect to the target variable. IT plays a vital role in this feature selection process, primarily through the concepts of entropy and IG [22]. Entropy, as defined in Equation (2), quantifies the impurity or randomness of a dataset. In the context of DT and RF, it measures the degree of disorder within a set of examples. A node with high entropy indicates a mixed distribution of classes, while a node with low entropy (or zero entropy) contains examples primarily from a single class. IG as defined in Equation (9), measures the reduction in

entropy achieved by partitioning the dataset based on a specific feature. It represents the amount of IG about the target variable by knowing the value of a particular attribute. The feature with the highest IG is chosen as the splitting attribute at each node because it provides the most information about the class labels, leading to the purest possible child nodes. DT and RF models iteratively select the feature with the highest IG to split the data at each node. This process continues recursively until a stopping criterion is met, such as reaching a maximum tree depth, having a minimum number of samples in a node, or achieving sufficiently low entropy in the leaf nodes.

Figure 6 shows an example of using entropy and IG in DT applied to the well-known Iris dataset. The tree employed entropy and IG to identify the optimal features for splitting the data, facilitating the classification of the flowers into their various species. The IG for each feature can be calculated to determine the best split at the root node. For example, "petal length" typically has a high IG, effectively separating the Iris *setosa* class from the other two. The algorithm also demonstrates the sequence of questions and decisions applied to each feature.

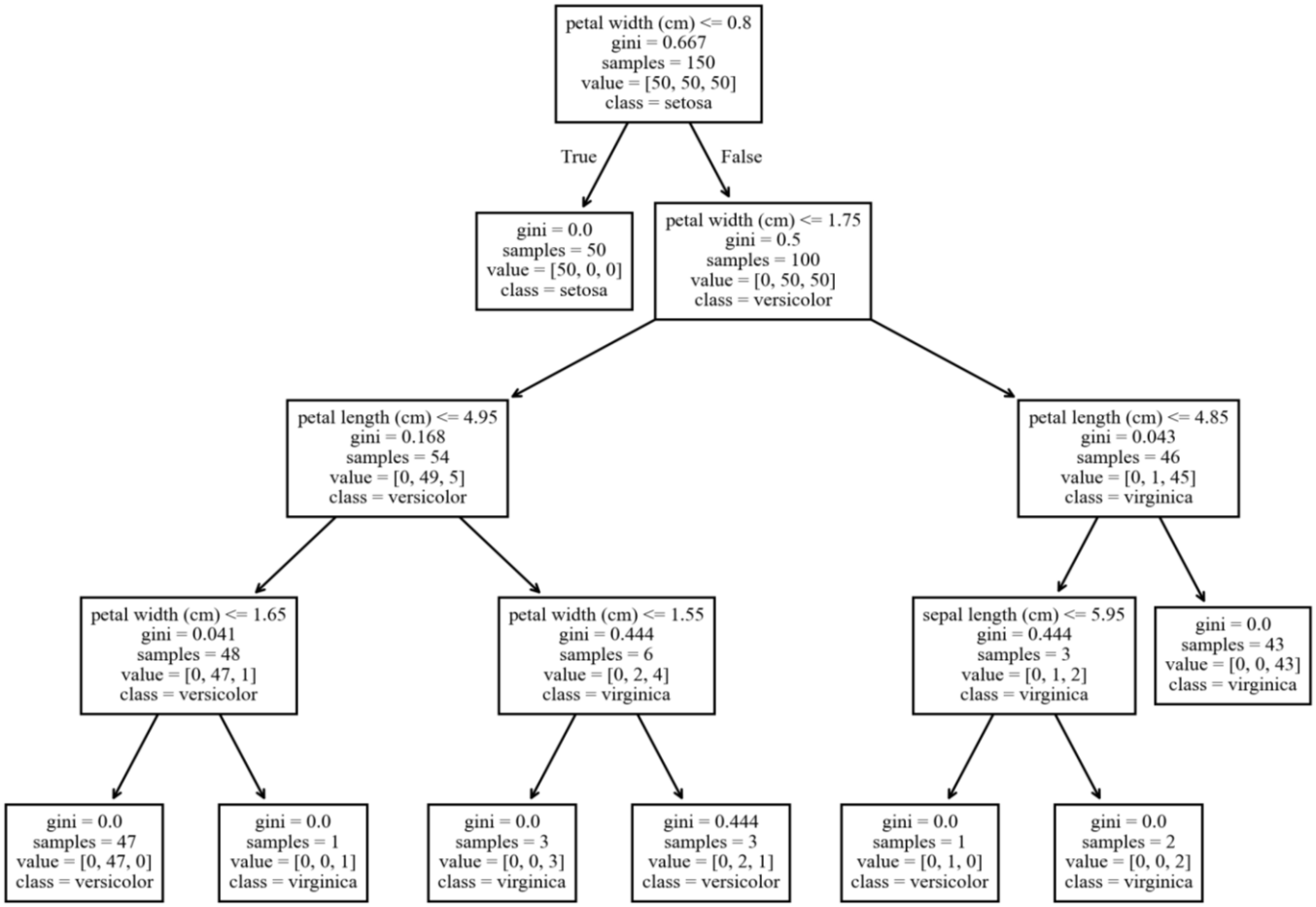


Figure 6. DT splitting using entropy and information gain.

In DT, IG has certain limitations. For instance, it may favor features with many unique values, as they often yield splits with lower entropy-even when those features are not meaningful. To mitigate this, the gain ratio can be used, which adjusts IG by the feature's intrinsic entropy. Additionally, IG can increase computational complexity, particularly for continuous features since discretization is required to calculate it. This process not only affects performance but may also introduce bias. On the other hand, boosted trees such as XGBoost take a different approach. While the underlying weak learners are still trees, the splitting criteria are often derived from a loss function optimization perspective, such as logistic loss or Mean Squared Error (MSE).

Table 1 shows the classification accuracies of both the DT and XGBoost utilizing different node splitting criteria. The table shows that utilizing entropy-based node splitting has achieved more accurate model than using MSE loss optimization function of the XGBoost model.

Table 1. DT vs XGBoost accuracy.

Model	Accuracy%
DT (entropy)	93
XGBoost (MSE)	91

4.2. Mutual Information in Feature Selection

MI, is a metric that evaluates the amount of information one random variable contains about another random variable. In the context of feature selection, it measures the correlation between the feature and the target variable. A higher MI value indicates that the feature is

more informative with respect to the target variable. What sets MI apart is its ability to handle both linear and non-linear relationships between variables, unlike other methods which are limited to linear relationships. This makes it a robust and adaptable approach to feature selection, allowing for the exploration of complex relationships between variables. By capturing both linear and non-linear dependencies, MI provides a comprehensive understanding of the data and enhances the accuracy and effectiveness of feature selection algorithms. Therefore, utilizing MI for feature selection can significantly contribute to the success and efficiency of data analysis. Additionally, it enables the identification of key features that have a significant impact on the target variable, allowing researchers and analysts to focus on the most relevant aspects of the data.

In summary, MI is a crucial component in the field of feature selection, offering valuable insights and empowering data-driven decision making [25]. A typical approach for feature selection utilizing MI is:

- Determining MI: evaluate the MI of each feature with the target variable.
- Prioritizing features: arrange the features according to their MI scores.
- Choosing features: select the top-rated features to construct the ML model.

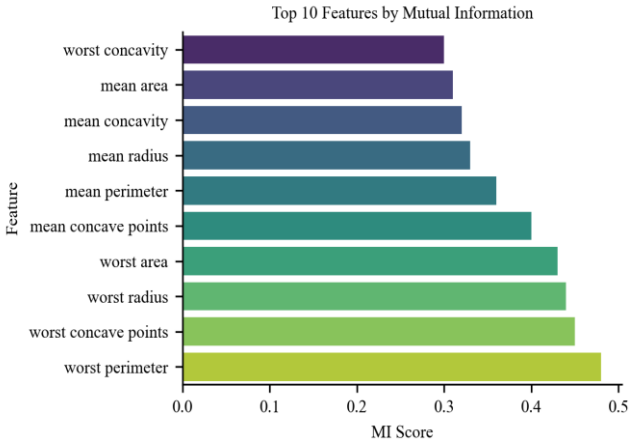


Figure 7. MI scores of breast cancer dataset features.

Figure 7 shows the MI scores ranked from highest to lowest for the renown breast cancer dataset which contains features computed from digitized images of breast mass and a target variable indicating the presence of cancer. The findings present the MI scores for each feature in the breast cancer dataset, showing the contribution of each feature to predicting the target (presence or absence of cancer). Higher MI scores indicate that a feature is more informative and pertinent to the classification task. Sorting these scores facilitates the identification of the most significant features, guiding feature selection for the development of more efficient models.

To compare the results, ANalysis Of VAriance (ANOVA) F-score is utilized to priorities features according to their F-scores as shown in Figure 8.

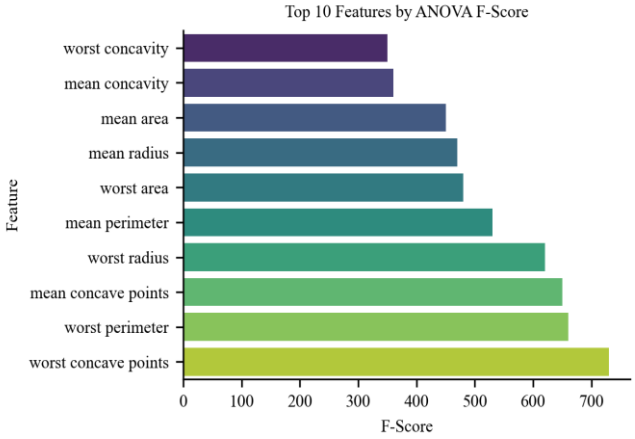


Figure 8. ANOVA F-scores of breast cancer dataset features.

Table 2 shows the classification accuracy summary when using logistic regression model trained using the top *k* features selected by each method.

Table 2. Classification accuracy summary.

Method	# of features (k)	Accuracy%
MI	5	93
F-score	5	92.3
MI	20	95.1
F-score	20	94.7

4.3. Cross-Entropy in Classification Models

In classification models like logistic regression and NNs, cross-entropy is frequently used as a loss function to measure the difference between the predicted probabilities and the true labels. The main goal is to effectively adjust the model's parameters to reduce this loss, thereby improving the accuracy and dependability of the predictions, and ultimately optimizing the overall performance and efficacy of the model [17]. For example, the cross-entropy loss for multi-class classification problems, where the target variable, y, might have values from a collection of classes $\{1, 2, \ldots, C\}$ is defined as:

$$L = -1/N \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log(p_{ic}) \quad (10)$$

where:

C is the number of different classes.
N is the number of samples.
y_{ic} is a binary variable (0 or 1) denoting whether class label c accurately classifies sample i.
p_{ic} is the predicted probability of sample i belonging to class c.

Figure 9 shows the role of cross-entropy as a loss function in neural network for multi-class classification tasks. The cross-entropy loss is used to calculate the error between the true and predicted outputs. This error is then propagated back through the network (as indicated by the dashed arrows labeled error back

propagation), adjusting the weights, w_{ij}, to minimize the loss. This iterative process continues until the model's predictions are optimized to match the true labels as closely as possible. In summary, cross-entropy loss is a recommended choice for training NNs on categorical outputs since it penalizes incorrect classifications harder than small errors in classification models. Reducing the cross-entropy loss helps the model to learn to produce predictions that closely correspond with actual class labels, therefore enhancing classification accuracy.

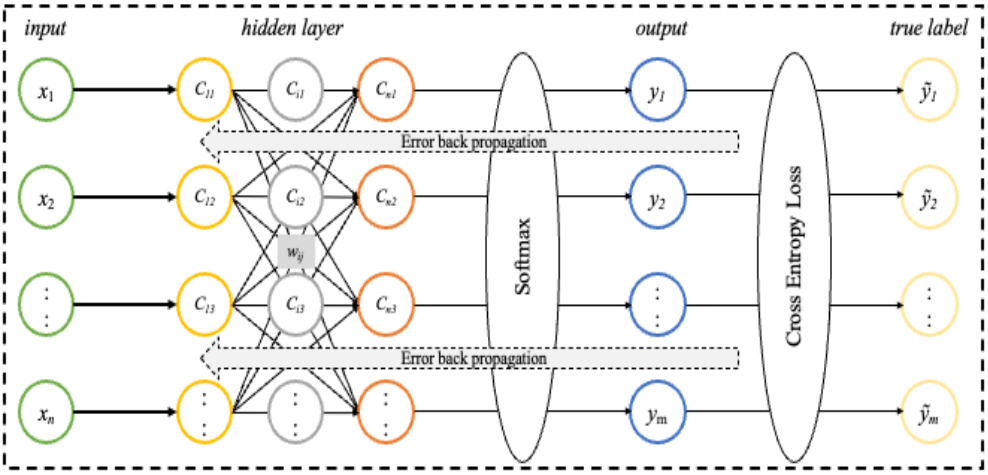


Figure 9. NN with cross-entropy as loss function [26].

Figure 10 illustrates how the cross-entropy loss decreases over the training epochs for both the training and validation sets utilizing the Iris dataset. Neural network model is used for the classification of the iris species. This example features a basic feedforward neural network comprising a single hidden layer containing 10 neurons. The hidden layer employs a ReLU activation function, appropriate for identifying non-linear relationships in the data, whilst the output layer utilizes the *softmax* activation function. The *softmax* layer is essential for multi-class classification tasks, as it transforms the raw output scores into probabilities for each class.

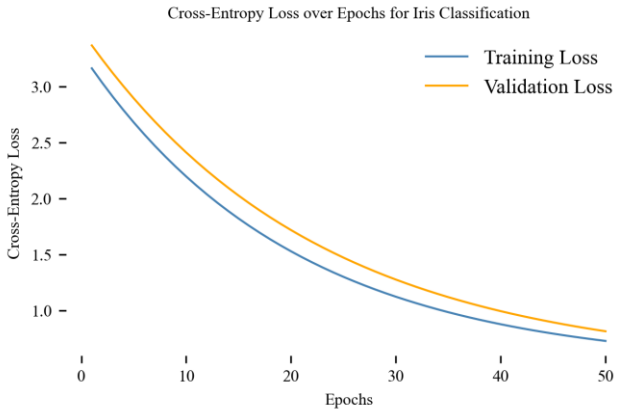


Figure 10. Cross-entropy loss for Iris classification.

A reduction in cross-entropy loss during the learning phase signifies that the model's predictions are increasingly accurate and align more closely with the true distribution. In an effectively trained model, both the training and validation losses should diminish and ultimately converge, indicating that the model is effectively learning and generalizing. An increase in validation loss concurrent with a drop-in training loss signifies overfitting, wherein the model is memorizing the training data and inadequately generalizing to unknown data.

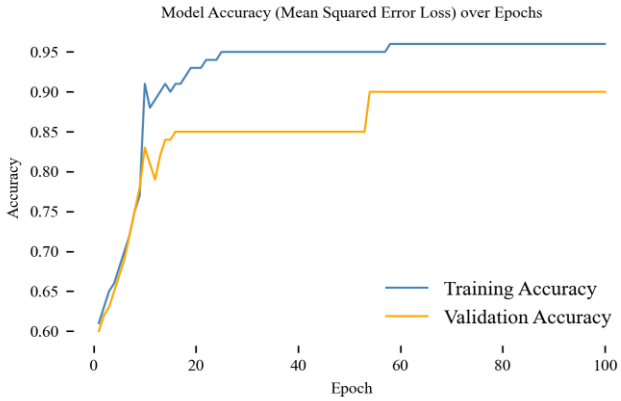


Figure 11. Model accuracy using MSE loss over epochs.

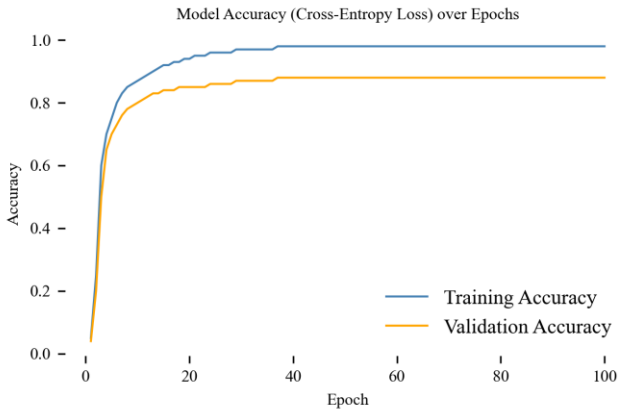


Figure 12. Model accuracy using CE loss over epochs.

To compare the result, a second experiment trains the

same model architecture using MSE loss function. While MSE is a common loss function, it's generally used for regression tasks. Applying it to classification serves as a non-IT centric comparison. It penalizes the squared difference between the predicted probabilities and the target values. Figures 11 and 12 depict training and validation accuracy over 100 epochs for models optimized using different loss functions: MSE and CE Loss, respectively. Figure 11 shows both training and validation accuracy start around 0.60 and gradually improve, with training accuracy reaching above 0.95 and validation stabilizing around 0.92. The learning curve shows occasional plateaus and minor fluctuations, suggesting a slower convergence and potential sensitivity to noise or suboptimal loss choice for classification. In contrast, Figure 12 exhibits a much faster and smoother convergence, with training accuracy quickly approaching 0.98 and validation accuracy closely following, plateauing near 0.95. The reduced gap between training and validation performance in the CE model indicates better generalization and more stable learning, making it more suitable for classification tasks compared to the model trained with MSE.

4.4. KL-Divergence in Variationa Autoencoders (VAEs)

VAEs are generative models designed to understand the underlying distribution of the data, see Figure 13. These models consist of an encoder that maps input data to a latent space and a decoder that maps the latent space back to the data space. Unlike traditional auto-encoders, VAEs impose a probabilistic structure on the latent space, enabling them to create new data points by sampling from the learned distribution. The incorporation of KL-divergence in VAEs serves to measure the distinction between the learned latent variable distribution and a prior distribution, typically a standard normal distribution. This regularization term guarantees that the latent space exhibits desirable characteristics, such as smoothness and continuity, which are crucial for generating new data points. The loss function of a VAE encompasses two components: *Reconstruction (Reconst) Loss*, which assesses the decoder's ability to reconstruct the input from the latent space, and KL-divergence Loss, which regularizes the latent space to align with the prior distribution [7].

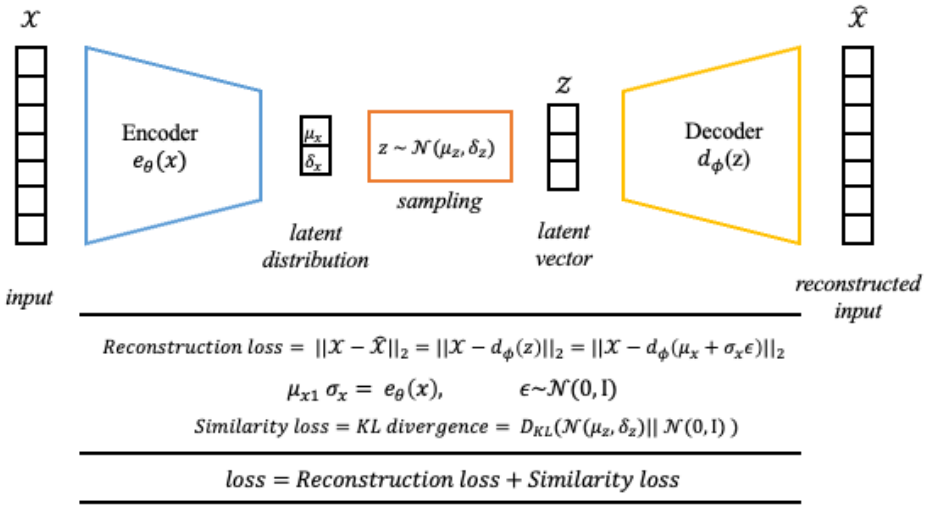


Figure 13. VAE with Kl-divergence.

The total VAE loss is given by:

$$Loss = Reconst\ Loss + \beta \times KL - Divergence \quad (11)$$

Where β is a weighting factor (often set to 1).

For example, assume you're trying to learn how to draw digits by looking at a variety of handwritten numerals. You begin by studying the basic forms and then attempting to duplicate them. This is comparable to how a VAE operates. It first learns the structure of the digits (encoder) before attempting to generate new ones (decoder). KL-divergence in VAEs assures that the space in which these digit shapes exist (latent space) is smooth and well-behaved. This means that comparable shapes are near together, allowing for a smooth transition from one shape to another. The VAE employs two styles of learning: It learns to rebuild the digits it has observed (reconstruction loss). It learns to arrange the shapes in a nice, smooth space (KL divergence). By integrating these two learning processes, the VAE can create new, realistic digits that resemble the ones it learned from.

To highlight the importance of KL-divergence in shaping the latent space in VAE, the VAE was built utilizing the Modified National Institute of Standards and Technology (MNIST) dataset. We compare the result with a standard Auto-Encoder (AE) that uses deterministic latent space.

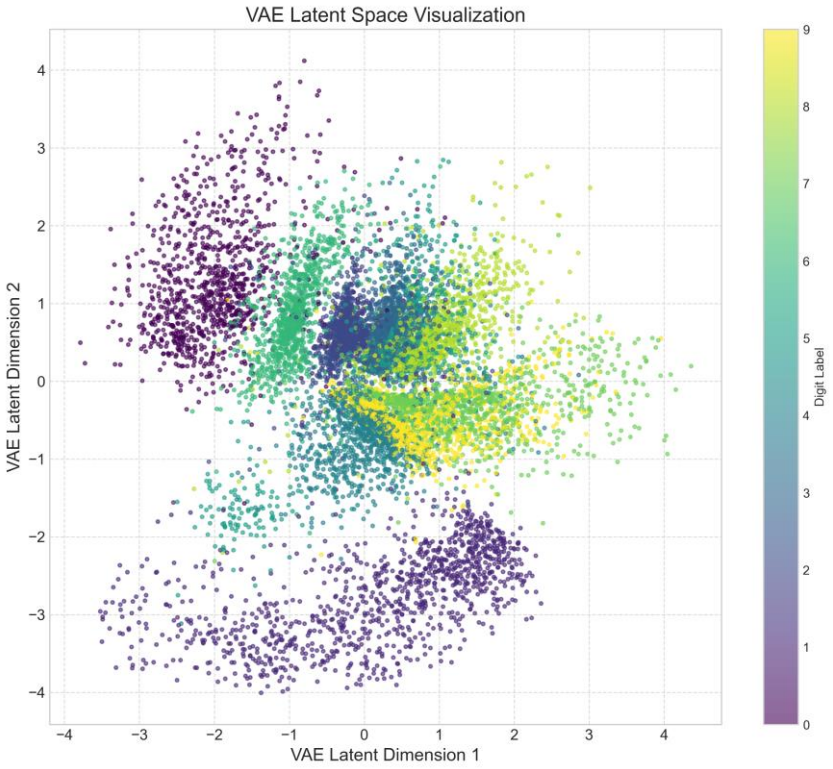


Figure 14. VAE latent space visualization.

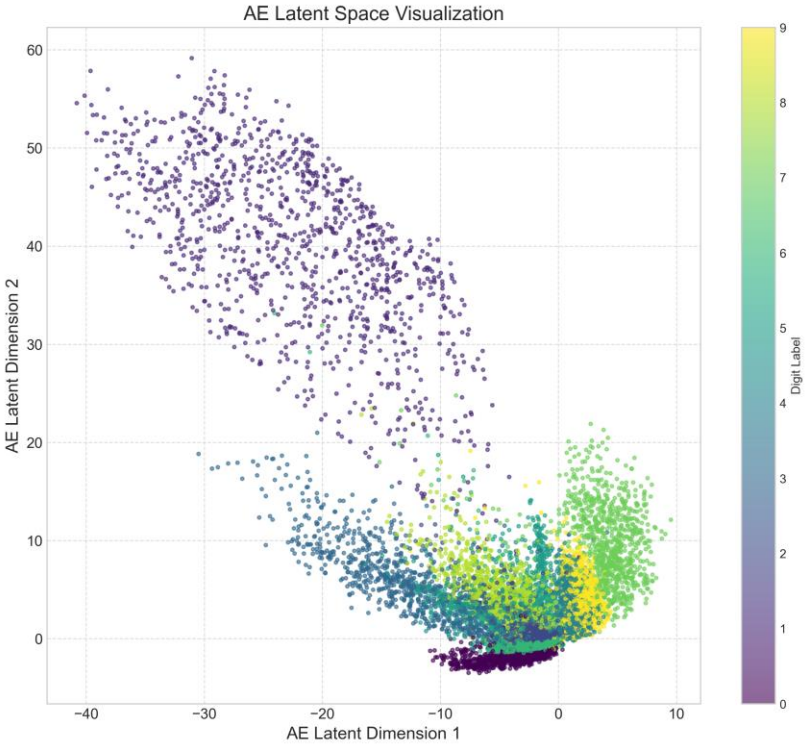


Figure 15. AE latent space visualization.

Figures 14 and 15 illustrate how a VAE and a standard AE represent MNIST digits in a reduced two-dimensional space. The VAE, which incorporates KL divergence as a regularization term, produces a more compact and continuous latent distribution, with smoother transitions between digit clusters and a more normalized structure across space. This regularization forces the encoded representations to approximate a prior distribution (typically Gaussian), supporting better interpolation and generative performance. In contrast, the AE yields tighter and more distinct clusters, particularly for certain digits, but exhibits a more irregular and stretched latent space. While the AE captures local structure effectively for reconstruction and class separation, it lacks the global organization and smoothness that characterize the VAE. Thus, the VAE provides a more generative-friendly encoding.

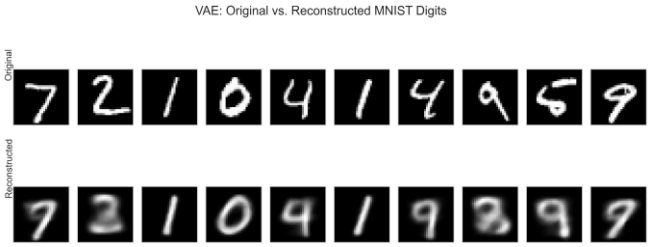


Figure 16. VAE original (top) and reconstructed (bottom) images.

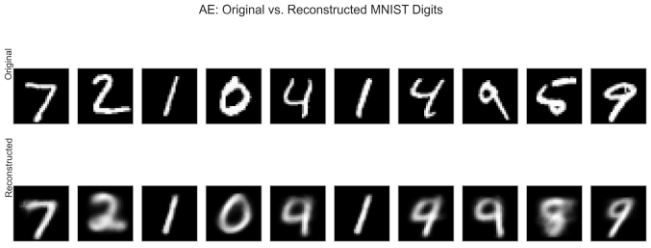


Figure 17. AE original (top) and reconstructed (bottom) images.

Figures 16 and 17 compare the reconstruction quality of MNIST digits by VAE and AE models. The AE, which uses a deterministic encoding-decoding process trained with MSE loss, produces reconstructions that are sharper and closer to the original digits, preserving fine structural details such as stroke width and contour precision. In contrast, the VAE employs a probabilistic latent space regulated by KL-divergence, leading to slightly blurrier reconstructions due to the inherent randomness introduced in sampling from the latent distribution. While the VAE's outputs are still recognizable and semantically faithful, they show a softer appearance with a minor loss in pixel-level clarity. This trade-off reflects the VAE's design focus on learning a smooth and continuous latent space suitable for generation and interpolation, whereas the AE prioritizes accurate one-to-one reconstruction of the input images.

4.5. Mutual Information and KL-Divergence in Information Bottleneck

Information Bottleneck (IB) principle provides a theoretical framework for learning representations that are optimally informative about the target (labels) but minimally informative about the raw input. IB attempts to compress data *X* into a representation *Z* that maintains only the information needed to predict the label *Y*. IB mathematically defines this as decreasing the MI between X and Z, I(X; Z), and maximizing the MI between Z and the labels Y, I(Z; Y). In practice, IB-based NNs frequently include a stochastic "bottleneck" layer and a regularization term that penalizes the network for encoding more information than necessary. In simpler terms, the IB aims to:

- Compress the input data into a compact representation (bottleneck).
- Preserve only the information that is relevant for predicting the output.

The IB is important because it provides a comprehensive theory for understanding how deep NNs learn and turn data into meaningful representations. It shows how networks continually reduce input while retaining the necessary characteristics for the task at hand. This viewpoint helps to explain why, despite their apparent ability to overfit, huge NNs frequently succeed in real-world scenarios: they focus on the most important traits and discard unnecessary detail over time. Furthermore, IB promotes interpretability by clearly outlining the trade-off between compression and relevance. It implies that effective representations-those that generalize well-find the correct balance between removing extraneous noise and maintaining task-critical information. This understanding explains why many over-parameterized models nonetheless perform well: they intuitively learn to focus on what is genuinely important while discarding unnecessary input that is irrelevant to prediction. To appreciate the importance of IB in classification models, Canadian Institute For Advanced Research (CIFAR-100) benchmark dataset is utilized. CIFAR-100 contains 60,000 color images, each sized 32×32 pixels. These images are divided into 100 different classes. Each class includes 600 images, 500 for training and 100 for testing. Because it covers such a wide variety of categories, it forces models to learn a broad range of visual features, making CIFAR-100 a good test of a model's generalization capability.

In our experiment, we contrasted a simple baseline Convolutional Neural Network (CNN) [4, 24] with an IB CNN. The baseline CNN only stacks convolution layers and trains by reducing cross-entropy losses. The IB CNN, on the other hand, uses a "bottleneck" a sophisticated process that attempts to compress the raw input into a smaller latent space while retaining the information required to accurately identify the images. This compression is aided by an extra penalty term (including KL-divergence), which allows the model to ignore noisy or unimportant information from the input data.

Figures 18 and 19 show the accuracy trends comparison between baseline CNN and IB CNN. The two models started around 12-13% accuracy, then rose steadily to reach around 55% for the baseline CNN and around 50% for the IB CNN in the training process. In the testing process IB CNN performance slightly outperforms the baseline CNN with a comparable result at epoch 50 (around 50% for the baseline CNN and around 51% for the IB CNN). IB approach, by design, places an additional regularization constraint via the KL-divergence term, which can slightly slow training progress early on but often help generalization. The experiment was repeated five times with different random initializations, to mitigate the impact of statistical fluctuations and provide a more robust and reliable assessment of the model's true generalization capabilities.

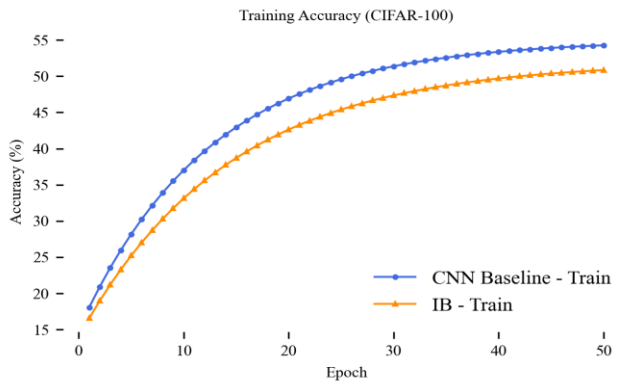


Figure 18. Training accuracy of IB and baseline CNN.

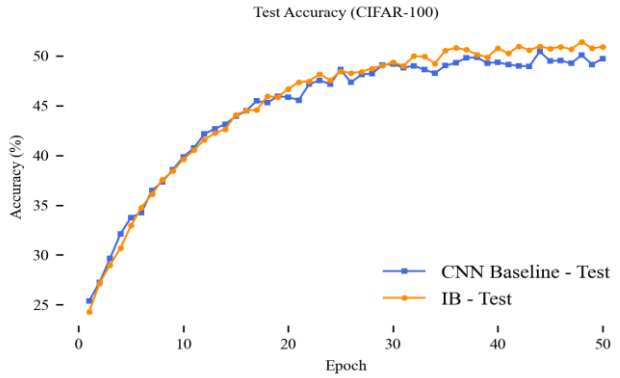


Figure 19. Testing accuracy of IB and baseline CNN.

For practical considerations: on CIFAR-100, a baseline CNN confronts classification issues, with 50% test accuracy after 50 epochs reflecting the dataset's complexity. IB CNN can provide even higher benefits in circumstances with noisy data, restricted labels, or longer training. In such cases, the model's emphasis on compressing unnecessary material frequently results in greater resilience and generalization than a baseline CNN.

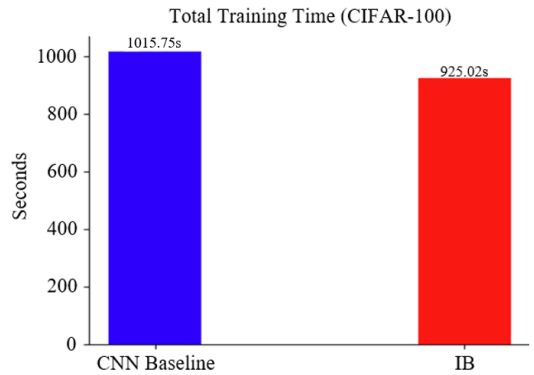


Figure 20. Total training time for the baseline CNN and IB CNN.

The experiment above was done on windows-based Laptop with intel core i9, with 32GB of RAM. The time taken for running the training process for the two models is shown in Figure 20. The figure shows that the baseline CNN model took about 1015.75 seconds to train, whereas the IB CNN model completed in around 925.02 seconds indicating that the overhead of the additional KL-divergence computations might not drastically increase training time.

5. Conclusions

This work systematically emphasized the critical importance of IT in both the theoretical foundations and practical applications of ML algorithms. Unpacking the basic IT measures-entropy, MI, cross-entropy, KL-divergence, and IG-made it clear how these constructs lead a variety of tasks, including feature selection, dimensionality reduction, and neural network training. For example, DTs use entropy and IG to reduce uncertainty at each node split, resulting in clear and understandable decision limits. NNs optimize cross-entropy to align predictions with ground-truth labels, while VAEs use KL-divergence to create a well-structured latent space that improves data creation and reconstruction quality. Furthermore, applying the IB concept demonstrates how NNs may manage the trade-off between efficient compression and high prediction fidelity, thereby improving both performance and interpretability.

From a broader perspective, empirical demonstrations on standard benchmarks-ranging from classification tasks on the Iris dataset to generative modeling with MNIST and feature-rich classification on CIFAR-100-show that IT-based metrics consistently improve generalization, robustness to noise, and model transparency. As ML applications become more complicated and data-driven, the solutions discussed here indicate that continuing to feed ML workflows with information-theoretic insights will be critical. Future extensions of this work may investigate more sophisticated architectures, multi-modal data scenarios, and advanced optimization schemes based on information-theoretic constraints, highlighting its critical role in shaping the next generation of interpretable and efficient ML systems.

Acknowledgment

The authors would like to thank the Arab Open University and Al-Zaytoonah University of Jordan for providing the necessary scientific research supplies to implement this work.

Funding

The authors extend their appreciation to the Arab Open University for funding this work through AOU research fund no. (AOUKSA-524008).

References

[1] Abril-Pla O., Andreani V., Carroll C., Dong L., and et al., "PyMC: A Modern and Comprehensive Probabilistic Programming Framework in Python," *Peer Journal Computer Science*, vol. 9, pp. 1-35, 2023. DOI:10.7717/peerj-cs.1516

[2] Algarni M. and Ben Ismael M., "A Dynamic Deep-Learning Approach for Predicting Information Diffusion," *International Journal of Advances in Soft Computing and its Applications*, vol. 15, no. 3, pp. 132-149, 2023. DOI:10.15849/IJASCA.231130.09

[3] Ali A., Naeem S., Anam S., and Ahmed M., "Shannon Entropy in Artificial Intelligence and its Applications Based on Information Theory," *Journal of Applied and Emerging Sciences*, vol. 13, no. 1, pp. 9-17, 2023. file:///C:/Users/acit2k/Downloads/549-1496-2-PB.pdf

[4] Alia M., Hnaif A., Alrawashdeh A., Jaradat Y., Masoud M., Manasrah A., and AlShanty A., "Robust Image Watermarking Using DWT, DCT, and PSO with CNN-Based Attack Evaluation," *The International Arab Journal of Information Technology*, vol. 21, no. 6, pp. 967-977, 2024. DOI:10.34028/iajit/21/6/1

[5] Alqudah A., Jaradat Y., Alobaydi B., Alqudah D., Alobaydi E., and Jarah B., "Artificial Intelligence in Design and Impact on Electronic Marketing in Companies," *Journal of Ecohumanism*, vol. 3, no. 4, pp. 170-179, 2024. DOI:10.62754/joe.v3i4.3480

[6] Alrumaidhi M., Farag M., and Rakha H., "Comparative Analysis of Parametric and Non-Parametric Data-Driven Models to Predict Road Crash Severity Among Elderly Drivers Using Synthetic Resampling Techniques," *Sustainability*, vol. 15, no. 13, pp. 1-30, 2023. https://doi.org/10.3390/su15139878

[7] Asperti A. and Trentin M., "Balancing Reconstruction Error and Kullback-Leibler Divergence in Variational Autoencoders," *IEEE Access*, vol. 8, pp. 199440-199448, 2020. DOI:10.1109/ACCESS.2020.3034828

[8] Avery J., *Information Theory and Evolution*, World Scientific, 2012. https://doi.org/10.1142/12668

[9] Bikku T., "Multi-Layered Deep Learning Perceptron Approach for Health Risk Prediction," *Journal of Big Data*, vol. 7, no. 1, pp. 1-14, 2020. https://doi.org/10.1186/s40537-020-00316-7

[10] Dargan S., Kumar M., Ayyagari M., and Kumar G., "A Survey of Deep Learning and its Applications: A New Paradigm to Machine Learning," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071-1092, 2020. https://doi.org/10.1007/s11831-019-09344-w

[11] Fan J., Li R., Zhang C., and Zou H., *Statistical Foundations of Data Science*, Chapman and Hall/CRC, 2020. https://doi.org/10.1201/9780429096280

[12] Jaradat Y., Masoud M., Jannoud I., Manasrah A., and Alia M., "A Tutorial on Singular Value Decomposition with Applications on Image Compression and Dimensionality Reduction," *in Proceedings of the International Conference on Information Technology*, Amman, pp. 769-772, 2021. DOI:10.1109/ICIT52682.2021.9491732

[13] Jeon H. and Roy B., "Information-Theoretic Foundations for Machine Learning," *arXiv Preprint*, vol. arXiv:2407.12288v4, pp. 1-96, 2024. https://doi.org/10.48550/arXiv.2407.12288

[14] Kanaan T., Kanaan G., Al-shalabi R., and Aldaaja A., "Offensive Language Detection in Social Networks for Arabic Language Using Clustering Techniques," *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 2, pp. 95-111, 2021. https://www.icsrs.org/Volumes/ijasca/2022.03.13.pdf

[15] Khder M. and Fujo S., "Applying Machine Learning-Supervised Learning Techniques for Tennis Players Dataset Analysis," *International Journal of Advances in Soft Computing and its Applications*, vol. 14, no. 3, pp. 190-214, 2022. DOI:10.15849/IJASCA.221128.13

[16] Mahesh B., "Machine Learning Algorithms-A Review," *International Journal of Science and Research*, vol. 9, no. 1, pp. 381-386, 2020. DOI:10.21275/ART20203995

[17] Mao A., Mohri M., and Zhong Y., "Cross-Entropy Loss Functions: Theoretical Analysis and Applications," *in Proceedings of the 40th International Conference on Machine Learning*, Honolulu, pp. 23803-23828, 2023. https://dl.acm.org/doi/10.5555/3618408.3619400

[18] Naskath J., Sivakamasundari G., and Begum A., "A Study on Different Deep Learning Algorithms Used in Deep Neural Nets: MLP, SOM, and DBN," *International Journal Wireless Personal Communications*, vol. 128, pp. 2913-2936, 2023. https://doi.org/10.1007/s11277-022-10079-4

[19] Ribeiro M., Henriques T., Castro L., Souto A., Antunes L., Santos C., and Teixeira A., "The Entropy Universe," *Entropy*, vol. 23, no. 2, pp. 1-35, 2021. https://doi.org/10.3390/e23020222

[20] Ruby U., Theerthagiri P., Jacob J., and Vamsidhar Y., "Binary Cross Entropy with Deep Learning Technique for Image Classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 13, no. 4, pp. 1-5, 2020. https://doi.org/10.30534/ijatcse/2020/175942020

[21] Sarker I., "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Computer Science*, vol. 2, pp. 1-21, 2021. https://doi.org/10.1007/s42979-021-00592-x

[22] Tangirala S., "Evaluating The Impact of GINI Index and Information Gain on Classification Using Decision Tree Classifier Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 1-8,

2020. DOI:10.14569/IJACSA.2020.0110277
[23] Vajda D., Pekar A., and Farkas K., "Towards Machine Learning-Based Anomaly Detection on Time-Series Data," *Infocommunications Journal*, vol. 13, no. 1, pp. 35-44, 2021. DOI:10.36244/ICJ.2021.1.5
[24] Zavalsız M., Alhajj S., Sailunaz K., Ozyer T., and Alhajj R., "A Comparative Study of Different Pre-Trained Deeplearning Models and Custom CNN for Pancreatic Tumor Detection," *The International Arab Journal of Information Technology*, vol. 20, no. 3, pp. 515-526, 2023. DOI:10.34028/iajit/20/3A/9
[25] Zhou H., Wang X., and Zhu R., "Feature Selection Based on Mutual Information with Correlation Coefficient," *Applied Intelligence*, vol. 52, pp. 5457-5474, 2022. https://doi.org/10.1007/s10489-021-02524-x
[26] Zhou Y., Wang X., Zhang M., Zhu J., Zheng R., and Wu Q., "MPCE: A Maximum Probability Based Cross Entropy Loss Function for Neural Network Classification," *IEEE Access*, vol. 7, pp. 146331-146341, 2019. DOI:10.1109/ACCESS.2019.2946264



Yousef Jaradat is an IEEE Senior Member and a Professor of Electrical and Computer Engineering at Al-Zaytoonah University of Jordan. He received his PhD from NMSU, New Mexico, USA, in 2012. His research interests include: Wireless Networks, Network Modeling and Simulation, AI and Machine Learning, Computer Security and Quantum Computing.



Mohammad Masoud is a Professor of Electrical Engineering at Al-Zaytoonah University of Jordan. He received his PhD in Communication Engineering and Information Systems from HUST, Wuhan, China in 2012. His research interests include: Computer Network Measurements, Network Security, Machine Learning, Software Defined Networking (SDN), Embedded Systems, Control Theory and Cyber Physical Systems (CPS).



Ahmad Manasrah is currently an Associate Professor of Mechanical Engineering at Al- Zaytoonah University of Jordan. He received his PhD degree from USF, Florida, USA. He was a Research Assistant and a member of Rehabilitation Engineering and Electromechanical Design Lab at the USF. He is also a member of ASHRAE, Jordan. His interests include: Renewable Energy, Smart Energy Technology Mechanical Control and Education.



Mohammad Alia is a Professor of Computer Science at AL-Zaytoonah University of Jordan. He received his PhD from USM, Penang, Malaysia, 2008. His research interests include: Public Key Cryptosystems, Fractals, Image Processing and Steganography, and Machine Learning.



Khalid Suwais is an Associate Professor in Computer Science. He received his PhD in Computer Science from University Sains Malaysia, Malaysia in 2009. Dr. Suwais is specialized in the field of Information Security and cryptography. His research interest includes: Cryptography, Information Security and Operational Research.



Sally Almanasra is an Associate Professor at the Faculty of Computer Studies, Arab Open University, Saudi Arabia. She received her PhD in AI and Software Engineering from Universiti Sains Malaysia in 2014. Her main teaching and research interests include AI, Information Security, and Game Theory.