# 检测技术设计

## 1.1. 整体架构设计

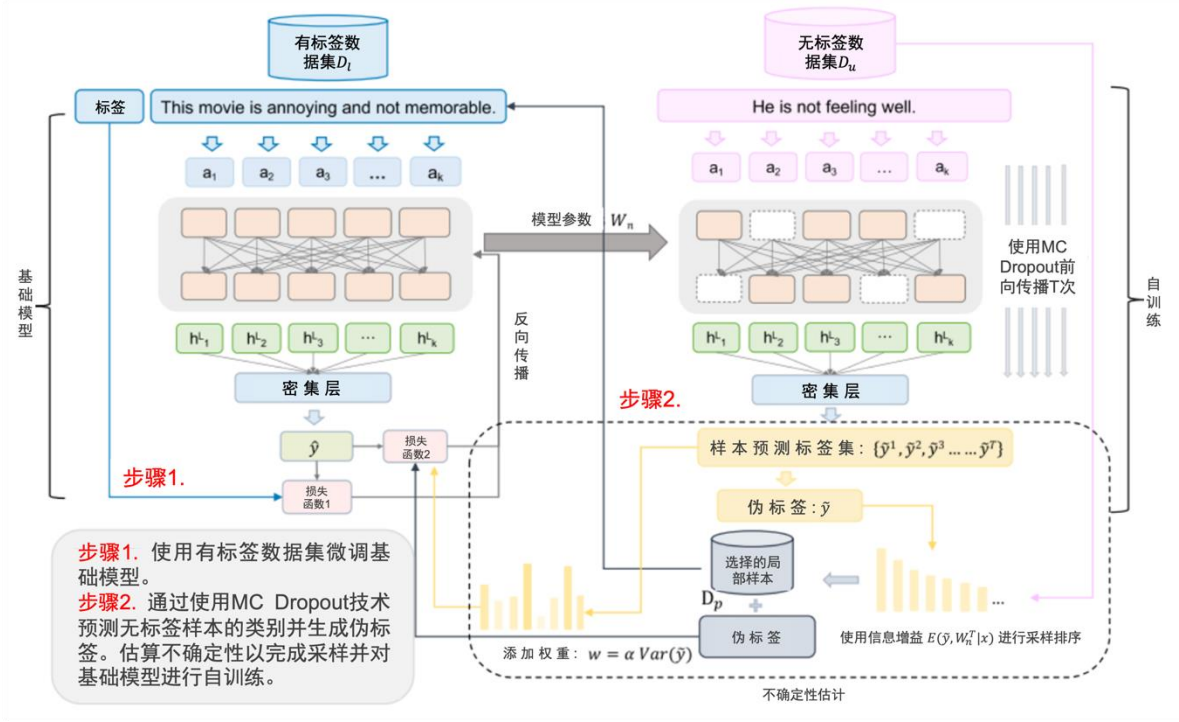


图1 整体技术架构

该检测系统主要运用人工智能技术，借助神经网络模型强大的特征提取与学习能力来完成恶意隐写载体检测任务。系统侧重于提取隐写分析特征，以发现隐写文本和普通文本分布之间的差异。如果方程（1）成立，那么我们的方法将文本 x 视为包含信息的隐写文本：

$$p(S \mid f(x)) > p(M \mid f(x)), \tag{1}$$

其中 S 表示隐写文本的类别，M 表示普通文本的类别，p 是学习得到的文本的后验概率分布，f(•) 是训练后的特征提取模块。

在实践中，可能会得到一个小的标记数据集 $D_l = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ 由 m 段文本 x 及其对应的标签 y 组成。通过网络爬虫[44]等方式，隐写分析器可以获得大型未标记数据集 $D_u = \{x_1, x_2, \ldots, x_z\}$ 由 z 段没有标签的文本 x 组成。这个小样本语言隐写分析任务是依靠非常小的标记数据集 $D_l$ 和辅助的未标记数据集 $D_u$ 来实现高效的隐写分析。

基于任务建模，该方法总体设计如下。其架构如图1所示，值得注意的是对于右侧自训练过程中的模型参数 $W_n^T$ ，在后续中可以看到随着 t 的变化，也就是 $W_n^1, W_n^2, \ldots, W_n^t, \ldots, W_n^T$ 各值是不同的，这并不是因为模型参数在训练过程中得到了更新，而是每个 $W_n^T$ 对应每次Dropout操作后产生的结果，整个过程并不涉及反向传播，模型参数更新。这也就是为什么推理阶段要比训练阶段快得多。具体来说，我们的系统由两部分组成：基础模型和自训练。我们主要利用强大的预训练语言模型BERT作为基础模型 $B^W(\cdot)$（其中 W 为模型参数）的特征提取器，它基于大量语料库进行训练，具有强大的文本特征表达能力。

在传统的使用场景中，Dropout操作在训练时被激活，在每个训练步骤中随机“丢弃”（即暂时移除）网络中的一部分神经元（或连接）。这样做可以迫使网络学习更加鲁棒的特征表示（减轻过拟合的风险），因为它不能依赖于任何一个神经元。

Yarin Gal和Zoubin Ghahramani于2016年提出MC Dropout方法[45]，他们证明了在测试时保持Dropout激活可以被视为贝叶斯近似，从而提供了关于神经网络预测的不确定性的信息。具体来说，通过在测试时保持Dropout激活，并对同一个输入进行多次前向传播，我们可以获得关于预测的分布，这可以被解释为模型不确定性的近似。

通过使用由少量有标记的隐写和正常样本组成的数据集 $D_l$ 对基础模型进行微调，基础模型可以提取和捕获初始的底层和简单的隐写分析特征。显然，少数样本的这些特征还远远不够，基础模型还没有达到较高的检测水平。然后我们利用自训练继续提取和学习更明显和通用的隐写分析特征。利用刚刚经过微调的基础模型 $B^{W_0}(\cdot)$ ，对 $D_u$ 中未标记的样本 x 进行预测，得到其伪标签 $\tilde{y}$ 。由于微调后的基础模型没有学习到足够的特征，过程中会存在较大的噪声和偏差，导致一些未标记样本的预测结果存在较大的不确定性。因此，我们利用不确定性估计来尽可能挑选出 $D_u$ 中更明显、更简单的样本，形成伪标记数据集 $D_p = \{(x_1, \tilde{y}_1), (x_2, \tilde{y}_2), \ldots\}$ 。$D_p$ 有利于后续的训练过程提取和学习足够多的、更明显的特征，逐渐明确分布差异。我们可以多次重复这个过程，然后构建出有效的小样本困境下的语言隐写载体分析系统。

## 1.2. 高维文本特征向量提取

在当前信息迅猛增长的背景下，文本数据的数量和复杂性均呈现急剧上涨的趋势，涵盖了从社交媒体、新闻稿件到学术论文和客户反馈等广泛领域。为了有效地解析和理解这些海量文本数据，高维文本特征向量提取作为一种关键的技术手段显得尤为不

可或缺。高维特征向量的应用能够捕捉文本数据中的复杂模式和深层次语义信息，这与传统的低维表示方法（如词袋模型和TF-IDF）形成鲜明对比，后者往往难以有效保留句子的上下文信息。通过采用如深度学习的词嵌入（word embeddings）和上下文化词表示(contextualized word representations),特别是基于BERT(Bidirectional Encoder Representations from Transformers）的模型，高维特征向量可以全面且细致地提取文本中的语义关联和潜在主题。

### 1.2.1. 句子嵌入Token生成

在使用BERT模型对文本数据集进行特征提取的过程中，生成Token并构建与BERT模型相对应的输入形式是处理文本数据的前提。BERT模型的输入通常包括三个主要部分：input_ids、attention_mask和token_type_ids。其生成步骤如下所述：

1. 加载BERT Tokenizer

首先，我们需要加载与我们所选择的BERT模型相对应的分词器（tokenizer）。在此系统中，我们选择使用bert-base-uncased作为我们的特征提取器。

2. 准备文本数据

之后，准备好需要处理的文本数据，在本作品中，这些文本来自多个社交平台，并使用两种不同的编码方法以评估模型的鲁棒性，所有文本数据被组织为一个列表，每个元素即为一个句子。

3. 使用Tokenizer进行处理

接下来，我们使用BERT的tokenizer对文本数据进行处理，包含几个重要步骤：

- **Tokenization**：将文本分割为一系列的子词（subwords），并将每个子词转换为其在词汇表中的索引。
- **添加特殊标记**：在每个文本序列的开始和结尾添加特殊的CLS标记（用于分类任务）和SEP标记（用于分隔句子）。
- **Padding和Truncation**：为了适应BERT模型的固定输入长度，需要将长短不一的句子序列填充（padding）到特定长度或截断（truncation）以适应BERT模型设定的最大输入长度（在此系统中为128个token）。
- **生成input_ids**：每个token被转换为在BERT词汇表中的唯一标识符，组成模型的输入序列。
- **生成attention_mask**：这是一个二进制向量，用于指示模型应该关注哪些

token。其中，1表示要关注的token（包括实际词语和特殊标记），0表示用于填充的token，不需要被模型关注。

- **生成token_type_ids**：用来区分多个句子的标记。在BERT模型中，句子对任务会用到这个矩阵，其中，第一个句子的token一般用0表示，第二个句子的token用1表示。

### 1.2.2. 正余弦奇偶位置编码

BERT模型的核心机制为多头自注意力机制，其自身并不具备位置感知能力，需要通过位置编码（Positional Encoding）来引入序列信息。BERT模型中的Transformer层采用了一种特殊的正余弦奇偶位置编码方法，以此来增强模型对序列顺序的感知能力，并提升其在文本处理中的应用效果。

该系统所使用的正余弦位置编码是一种巧妙且有效的方式，其核心思想是为序列中的每个位置生成一个唯一的向量，以此来表示其位置信息。具体而言，给定一个位置 pos 和一个维度 i，位置编码向量的元素通过公式 (2) 生成：

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \end{aligned} \tag{2}$$

其中，$d_{model}$表示模型的维度。该设计的独特之处在于，通过对不同位置和维度的组合使用正弦和余弦函数，模型能够获得位置间的相对关系。同时，由于正余弦函数具有周期性，位置编码向量可以生成足够丰富和唯一的表示，即使在较长的序列中也能保持有效区分。

在文本处理中，通过这样的正余弦奇偶位置编码，多头自注意力机制能够捕获文本中更为细腻的位置特征，位置编码向量被直接添加到词向量上，使得输入到Transformer中的词不仅包含其语义信息，还包含其在句子中的位置信息。这种方式确保了模型在处理序列数据时，可以利用位置信息来理解词语之间的顺序和距离，进而提升对上下文的捕捉能力。同时，所提供的位置信息为相对位置信息，这可使得模型可以更好地理解和保留长距离依赖关系，因为位置编码向量使得位置之间的相对关系被显式地编码在输入的特征向量中，这种特性使得我们的系统在处理复杂句子结构和长文本时，能够充分利用上下文信息，提升理解精度和检测准确性。

### 1.2.3. Transformer Encoder多层堆叠计算

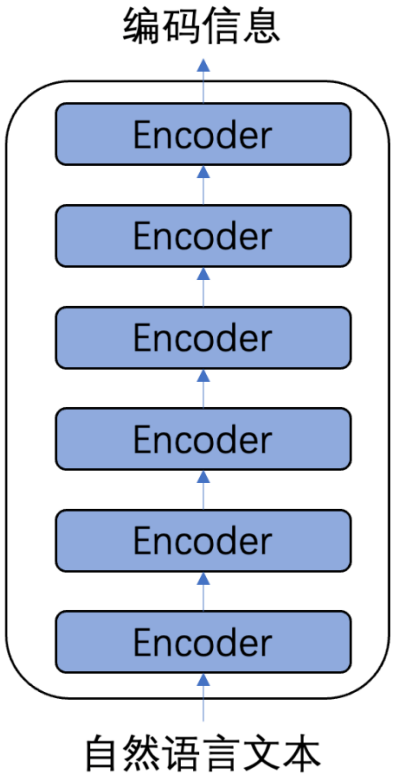


图 2 Encoder Layer 多层堆叠

Transformer Encoder 由多个相似的编码器层（Encoder Layer）堆叠而成，每个编码器层又由几个关键子组件构成，包括多头自注意力机制（Multi-Head Self-Attention Mechanism）、前馈神经网络（Feed-Forward Neural Network, FFN）以及 Add & Norm 层（残差连接和层归一化）。这些子组件协同作用，逐步提取和融合输入序列的语义信息，架构如图 2 所示，整体计算流程如下所述：

1. 输入嵌入与位置编码

在开始处理之前，输入的词语首先会被转换为词向量（Embeddings），并添加位置编码（Positional Encoding），以使模型能够感知词语的位置顺序。这些嵌入向量作为初始输入，传递给第一个编码器层。

2. 多头自注意力机制

每个编码器层的第一步是多头自注意力机制。自注意力机制可以看作是计算输入序列中每个词对所有其他词的关注程度。其计算步骤如下：

- **生成查询（Query）、键（Key）和值（Value）矩阵：**

从输入嵌入向量中通过不同的线性变换得到查询矩阵 Q、键矩阵 K 和值矩阵 V：

$$Q = XW^Q, K = XW^K, V = XW^V, \tag{3}$$

其中 X 为输入嵌入向量，$W^Q, W^K, W^V$ 为可训练的参数矩阵。

- **自注意力得分计算：**

进行点积注意力计算，通过点积 Q 和 K，然后经过缩放和 Softmax 操作，得到注意力权重矩阵 Attention(Q,K,V)：

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (4)$$

计算模块如图 3 所示。

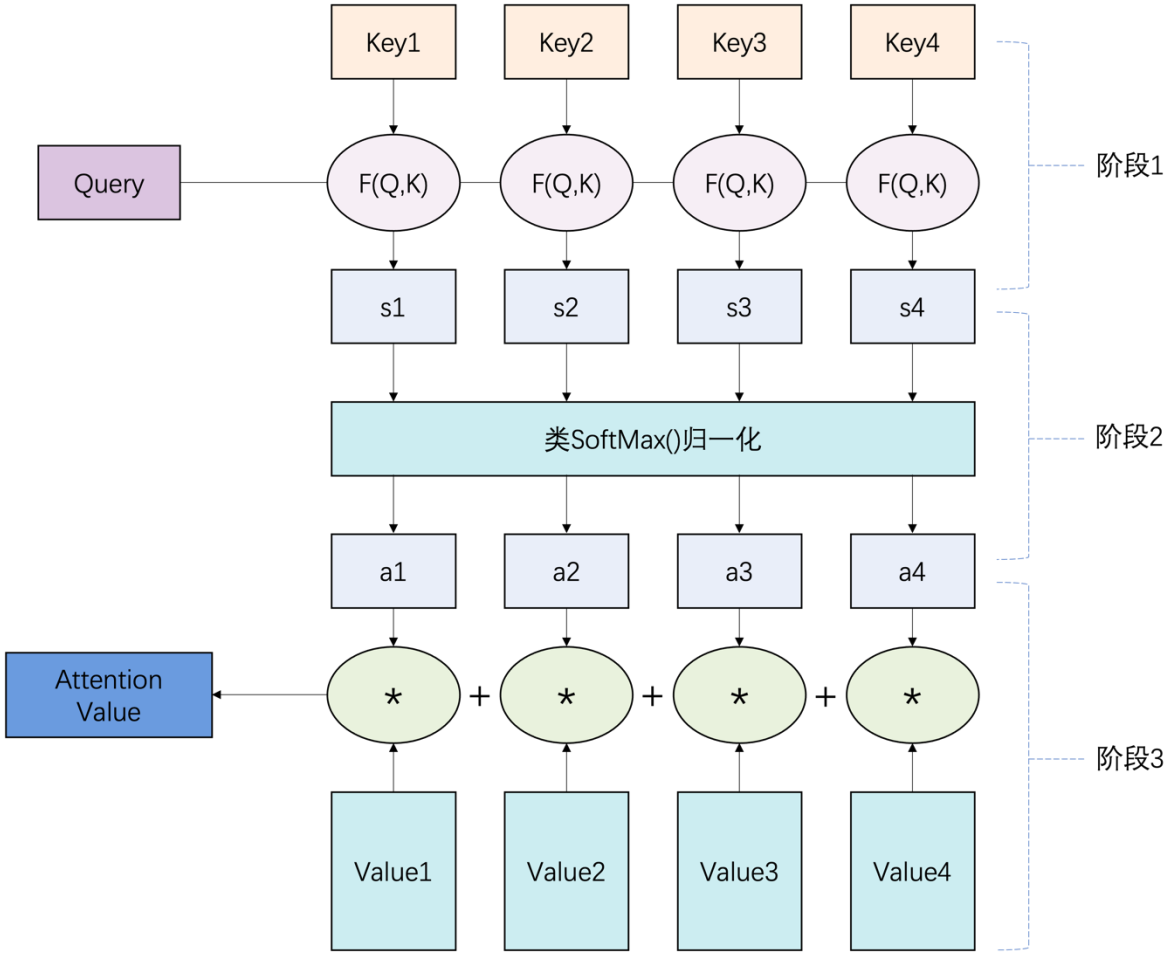


图 3 自注意力得分计算模块

- **多头注意力：**

为了捕捉更多的特征，每个编码器层会并行计算多个自注意力机制（称为多头注意力），并将这些头的输出拼接起来，再通过线性变换得到最终的多头注意力输出：

$$MultiHead(Q,K,V) = Concat(head_1, head_2, \ldots, head_h)W^O, \quad (5)$$

其中每个头 $head_i$ 的计算方式与单头注意力相同，架构如图 4 所示。

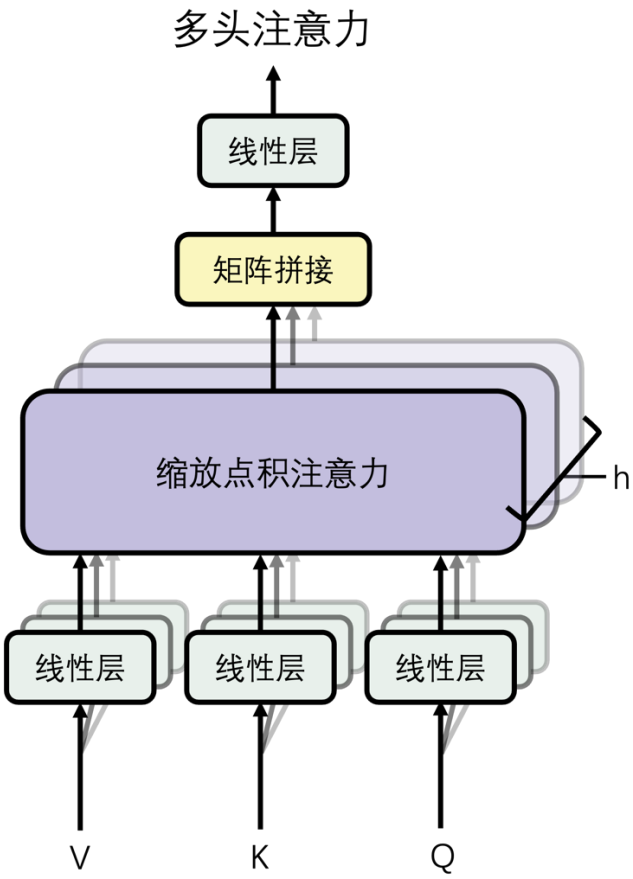


图 4 多头注意力架构

3. Add & Norm 层

多头自注意力机制输出被传递到一个 Add & Norm 层，首先进行残差连接（Residual Connection）[46]，即将该输出与输入相加，然后进行层归一化（Layer Normalization）[47]，以稳定训练过程并加速收敛：

$$Output = LayerNorm(X + MultiHead(Q, K, V))。 \quad (6)$$

4. 前馈神经网络（FFN）

上述的输出之后会通过一个前馈神经网络。FFN 是一个两层的完全连接网络，应用 ReLU 激活函数：

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2, \quad (7)$$

此处的参数 $W_1$, $b_1$, $W_2$, $b_2$ 都是可训练的矩阵和向量。

5. 再次通过 Add & Norm 层

在 FFN 处理之后，其输出会再次经过一个 Add & Norm 层，即进行残差链接和层归一化。

6. 多层堆叠

上述步骤构成了一个编码器层。Transformer Encoder 由多个这样的编码器层堆叠而成，每一层都会对输入进行更深层次的特征提取和语义融合。

通过将 Encooder Layer 多层堆叠来计算文本特征向量可以使得系统捕捉深层次的语义信息，增强表达能力，并通过渐进式的学习模式逐步聚合和筛选特征信息，提高最终的特征精度。

整体来说，BERT 模型中 Transformer 层的计算方式为：

$$\begin{aligned} H^0 &= E, \\ H^l &= Transformer_l(H^{l-1}), 1 \le l \le L, \end{aligned} \tag{8}$$

其中 $H^l$ 表示第 l 个 Transformer 层的输出，L 是我们使用的模型中的层数。

## 1.3. 基于贝叶斯近似的边缘似然估计方法

在统计学和机器学习中，贝叶斯方法提供了一种从数据中推断模型参数的系统方法，而边缘似然估计（Marginal Likelihood Estimation）则是关键的一步。边缘似然，也称为证据（Evidence），是贝叶斯模型比较和选择中至关重要的部分。然而，直接计算边缘似然通常是很困难的，这时基于贝叶斯近似的方法就派上了用场。

在贝叶斯推断中，我们希望通过数据 D 更新我们对模型参数 θ 的信念。贝叶斯定理提供了这一更新的基础：

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \tag{9}$$

其中，$p(D|\theta)$ 是似然函数（Likelihood），$p(\theta)$ 是先验分布（Prior），而 $p(\theta|D)$ 是后验分布（Posterior）。边缘似然（或证据）$p(D)$ 是通过积分对所有可能的参数 θ 融合的：

$$p(D) = \int p(D|\theta)p(\theta)\, d\theta, \tag{10}$$

边缘似然 $p(D)$ 在模型比较中扮演着重要角色，因为它量化了数据对于整个模型的支持程度。然而，直接计算这个积分在很多情形下是不可能的，特别是当参数空间很大或复杂时，因此我们需要对边缘似然作近似求解。

基于贝叶斯近似的边缘似然估计方法提供了一种强大而灵活的工具，用于复杂的贝叶斯推断问题。考虑到泛化性能和不确定性量化两个方面，系统最终选用 MC Dropout 来实现变分推理。

### 1.3.1. 选取易处理的模型参数先验分布

在贝叶斯框架中，先验分布是对模型参数的初始信念的数学表达。不同的先验分布可以反映我们对参数不同程度的信任和认知。例如，在完全不确定的情况下，我们将选择一个非常平的（即信息量很少的）先验分布；在有充足的先验知识的情况下，我们可能选择一个能够体现这种确切的信念，较为集中的先验分布。选择一个易处理的先验分布非常重要，因为它直接影响后验分布的计算和推断的复杂性。易处理的先验分布一般具有以下特性：

- **解析性强**：在计算后验分布时，解析性强的先验分布可以简化计算过程。例如，高斯（正态）分布和共轭分布。
- **计算效率高**：易处理的先验分布更容易通过数值方法进行计算，尤其是在参数空间较大时。

基于上述的特性，在系统中我们考虑使我们选择的分布向以下三种分布做近似：

1. 高斯（正态）分布

高斯分布是一种常用的先验分布，特别适用于线性模型和高维参数空间中。其解析性和对称性使其成为贝叶斯推理中的一个理想选择。其优点为：在大多数情况下，高斯分布具有良好的数学性质，易于进行解析计算。使用高斯分布通常能简化参数的优化过程。同时，当数据服从高斯分布时，后验分布往往也会保持高斯形式。

2. 共轭分布

共轭分布是指在某种特定的统计模型下，先验分布和后验分布属于同一分布族。这种特性使得更新后验分布的计算变得相对简便。在使用共轭分布时，更新后的后验分布保持与先验分布相同的形式，只需更新分布的参数即可，因此这一特性使得计算和推断过程得到了大幅度简化。

3. 均匀分布

均匀分布表示对参数没有任何偏好，给每个参数值相同的权重。它适用于完全不确定或没有先验知识的情况，该分布简单明了，适用于初始阶段以及对参数完全没有先验认识的情况。

考虑到数据充分性：数据的量和质量对先验分布的选择有重要影响，在此任务中即数据稀少的情况下，先验分布的选择显得尤为重要，应该尽量利用相关的先验知识。在模型健壮性方面：我们在选择先验分布时，也考虑了其对模型健壮性的影响。某些情况下，不适当的先验选择会导致模型结果不稳定甚至不可收敛，因此，在选择易处

理的先验分布时，我们也权衡了其对模型鲁棒性的影响。接下来我们通过以下若干步骤讲述将 MC Dropout 应用于深度神经网络时选择先验分布的多方考虑：

1. 确定模型参数：在深度神经网络中，参数主要是各层的权重和偏置项。

2. 分析数据特征：在此系统中，我们的任务是自然语言处理，因此对于某些层的权重，可以假设其初始值围绕零均值对称分布。

3. 评估先验分布：选择正态分布作为先验，因为它在权重和偏置项的处理上具有良好的解析性。在多维空间中，正态分布仍能保持较低的计算复杂度。

4. 共轭分布考虑：在某些线性层，正态分布与高斯噪音模型呈共轭关系，这进一步简化了贝叶斯推理的计算。

5. 参数选择：根据初步实验和经验，选择正态分布的均值为 0，方差为较小的正值（如 0.1），以反映对初始权重的初步认知和不确定性。

6. 验证与调整：通过初步训练和测试结果，验证选择的先验分布是否合适，若模型表现不理想，则调整正态分布的方差。

在应用 MC Dropout 进行变分推理时，此系统选择高斯分布作为神经网络权重和偏置项的先验分布，因为高斯分布具有良好的数学性质和解析性，这使得我们在对参数进行后验推断时能够简化计算复杂度；特别是在高维参数空间中，高斯分布的对称性和集中性能够有效地描绘我们对参数初始分布的合理预期，同时由于高斯分布与许多常见的噪声模型（如高斯噪声）呈共轭关系，这进一步简化了贝叶斯更新过程，使得神经网络的训练和测试阶段更为高效；此外，在 MC Dropout 框架下，我们需要在测试阶段进行多次前向传播，这相当于从先验分布中多次采样，而高斯分布固有的数值稳定性和易于采样的特性使其在这一过程中显得尤为适用，确保我们的蒙特卡罗采样过程能够顺利进行并提供可靠的不确定性量化，从而使得我们可以对模型预测结果和不确定性进行较为精确的衡量，大大提高系统的检测效果。

### 1.3.2. 基于观测样本对的后验分布建模

接下来，我们需要基于观测样本对进行后验分布建模。这个过程可以分为几个步骤，每一步骤都需要考虑理论和实践之间的平衡，并且需要充分利用之前选择的先验分布来指导后验更新，具体见以下九个步骤。

1. 初始化模型

首先，我们需要初始化神经网络模型的参数，通常是权重和偏置项。在我们选择

了高斯分布作为这些参数的先验分布后，我们会根据高斯分布的统计特性为每个参数赋初值。具体来说，权重和偏置项通常被初始化为零均值并具有某个小的方差。这种初始化方法不仅符合高斯分布的对称性，也有助于训练初期的稳定性。

2. 引入观测数据

通过观测数据样本对（比如输入数据和对应的标签），我们开始进行具体的模型训练。这里的观测数据不仅包括训练数据集的所有特征变量，同时也涵盖了对应的目标值或标签，这些数据点将指导后验分布的更新。通过反复观察和利用这些样本对，我们能够在训练过程中逐步更新模型参数，从而使模型更好地拟合数据。

3. 前向传播与 Dropout 操作

在训练阶段，每次前向传播时，我们会应用 Dropout 操作，这意味着我们随机地以一定概率丢弃网络中的一些节点，实际上是对网络进行子集采样。每一次 Dropout 操作生成不同的子网络，这样可以看作是从参数的近似后验分布中采样。具体来说，每个神经元以某一固定概率被保留，未被保留的神经元在当前迭代中不参与计算。

4. 计算损失并进行后向传播

在每次前向传播生成的子网络上，我们通过计算损失函数（通常是交叉熵损失或均方误差损失）来评估模型的预测效果。损失函数度量了模型预测值与真实观测值之间的差异。通过后向传播算法，我们计算损失函数对于神经网络参数的梯度，并利用这些梯度更新模型参数。

5. 参数更新与优化

在参数更新时，使用优化算法（此系统中采用 Adam 优化器）通过反复迭代最小化损失函数，从而逐步调整模型参数，使其更好地拟合数据。由于我们使用的是高斯先验分布，这些参数的更新过程实际上是对其后验分布的逐步演化，使得每次参数更新后的分布更接近于观测数据可能对应的真实参数分布。

6. 多次采样与不确定性量化

在训练完成并进入测试阶段后，我们仍然继续应用 Dropout 操作，通过多次前向传播来生成一系列可能的预测结果。这实际上是从模型参数的近似后验分布中多次采样，并通过这些采样结果来计算预测值的均值和方差。这些均值和方差不仅给出了模型的预测值，也提供了预测结果的不确定性量化。基于这些量化结果，能够更好地理解模型在不同输入条件下的可靠性和稳定性，从而为后续的决策提供依据。

7. 后验分布更新与模型评估

通过不断引入新的观测数据并重复上述步骤，模型的后验分布会不断更新，使其越来越准确地描述实际情况。在每一轮新的数据引入后，通过 MC Dropout 对新的观测样本进行回顾性分析，更新并记录参数的后验分布。同时，通过交叉验证或留出验证集评估模型性能，我们可以判断模型是否有过拟合或欠拟合的倾向，以便调整模型结构或先验分布参数。

8. 选择与优化先验分布参数

随着观测样本数据的逐步引入，我们会不断优化先验分布的参数，例如高斯分布的均值和方差。最初的先验选择可能不足以完美描述参数空间，因此在后验分布建模过程中，适时调整先验分布参数能大幅增强模型的拟合能力。这个过程需要结合模型在验证集上的表现以及对后验分布变化趋势的理解。

9. 模型不确定性分析与解释

最终，通过对模型不确定性的量化与分析，我们不仅获得了具体的预测结果，还能够理解和解释模型的行为。例如，通过分析不同输入情况下预测方差的变化，我们能识别出模型在某些输入区域的不确定性是否过高，进而改善这些区域的输入特征，或合理处理预测结果。

通过以上步骤，我们可以在对后验分布参数更新的同时，完成对后验分布的建模。在我们的系统中，贝叶斯神经网络 (BNN) 是通过对网络参数的概率分布（表示为 W）进行推断来实现的。具体来说，可以通过假设来自易处理族的先验分布 $q_\theta(W)$ 来对具有训练输入目标对 (x, y) 的参数空间上的后验分布 $p(W|X,Y)$ 进行建模。那么我们可以将贝叶斯定理[48]应用到基本模型 $B^W(\cdot)$ 上：

$$p(W|x,y)=\frac{p(y|B^W(x))q_\Theta(W)}{p(y|x)}\propto p(y|B^W(x))q_\Theta(W), \tag{11}$$

在上述公式中，$q_\theta(W)$ 表示 Dropout 分布，即在贝叶斯神经网络中，网络权重不是一个固定值，而是一个分布，因此我们可以估算模型的不确定性，并以此来完成对后验分布的更新与建模。

### 1.3.3. 蒙特卡罗Dropout变分推理

蒙特卡罗 Dropout 变分推理是一种利用神经网络进行贝叶斯推理的技术，特别适用于深度学习中的不确定性估计和边缘似然的近似计算。

蒙特卡罗 Dropout 来自 Dropout 正则化技术[49]，这是一种在深度学习中广泛应用的防止过拟合的技术。在训练过程中，Dropout 会随机丢弃一些神经元，让模型在多个不同的子网络上学习。这种技术不仅提高了模型的泛化能力，还可以通过多次前向传播评估不确定性，为我们将 Dropout 技术引入变分推理提供理论基础。在贝叶斯推理中，我们关注的是给定数据时模型参数的后验分布。然而，由于深度学习模型参数空间的庞大，我们无法直接精确计算这种分布。变分推理正是为此提供了一种有效的近似方法。传统的变分推理会选择一种特定的近似分布来逼近真实的后验分布，然后通过优化算法来调整这两者之间的距离。在蒙特卡罗 Dropout 中，我们使用 Dropout 生成多个不同的子网络，来构造这种逼近分布。

具体而言，在推理阶段，我们利用 Dropout 对神经网络参数进行随机丢弃，生成许多不同的网络配置。这些配置实际上代表不同的参数样本，这样我们就得到了一个从后验分布中采样的近似分布，之后我们通过对多次前向传播的结果进行平均及方差计算，实现对模型预测的均值和不确定性的估计。

接下来，我们探讨计算边缘似然的方法。边缘似然指的是数据在给定模型结构下的概率分布，而不是给定特定模型参数下的数据概率。蒙特卡罗 Dropout 变分推理的一个优势就是它可以使用多次采样来近似边缘似然。具体过程是这样的，我们首先对模型进行多次前向传播，每次前向传播都相当于从模型的后验分布中采样一个参数组合。然后我们将多次前向传播的概率结果进行平均，这个平均值即为数据的边缘似然估计。这个过程背后的理论基础是变分推理和蒙特卡罗方法的结合。变分推理提供了一个框架，将复杂的后验分布问题化简为优化问题；而蒙特卡罗方法则通过大量的样本计算来近似期望值。在蒙特卡罗 Dropout 中，通过局部的随机丢弃，我们生成了一个从近似后验分布中采样的过程，从而无须明确表示这个近似分布。

蒙特卡罗 Dropout 方法本质上是一种近似方法，因此会有一定的不精确性，但是这种不精确性在实践中往往是可以接受的，特别是在处理大规模数据和复杂网络时，它的效率和简便性使得它成为一种非常有吸引力的方法。此外，Dropout 还具有天然的正则化优势，这对于提高模型泛化性能也是有利的。

在总结多次前向传播的结果时，我们不仅可以得到对模型预测结果的期望值，还可以通过样本间的方差估计来衡量预测的不确定性。蒙特卡罗 Dropout 变分推理通过结合变分推理的优化思路和蒙特卡罗方法的样本计算，使得我们能够有效地近似计算

深度学习模型中的边缘似然。在具体操作中，我们利用 Dropout 技术生成多个模型变体，通过多次前向传播进行概率计算。这种技术不仅提供了对模型参数后验分布的近似，还能有效地评估模型预测的不确定性，为深度学习模型提供了更丰富的信息。

在我们的系统中，假设 Dropout 操作有 T 次，我们就可以获得 T 个屏蔽模型权重：$\{W^t\}_{(t=1)}^{T} \sim q_\Theta(W)$。“屏蔽”的意思是：掩盖掉部分神经元，也就是 Dropout 操作。分类任务中结果的概率 $p(y=c|x)$（c 为种类，例如 0 或 1）现在可以使用贝叶斯推理[50]通过蒙特卡罗积分获得：

$$\begin{aligned} p(y=c|x) &= \int_W p(y=c|\mathcal{B}^W(x))q_\Theta(W)dW \\ &\approx p(y=c|\mathcal{B}^W(x))q_\Theta(W)dW \\ &\approx \frac{1}{T}\sum_{t=1}^{T} p(y=c|\mathcal{B}^{W_t}(x)). \end{aligned} \quad , \tag{12}$$

## 1.4. 二元分类器设计及伪标签计算

为了设计二元分类器并计算伪标签，我们需要从多角度考虑一些常见的二元分类器方法，包括逻辑回归（Logistic Regression）、支持向量机（SVM）、决策树、随机森林（Random Forest）、以及密集层加 softmax 的神经网络分类器，但最终我们选择在最后一种分类器以应用 MC Dropout 技术实现伪标签计算，即密集层加 Softmax 的神经网络分类器，如下文所述：

神经网络是基于人工神经元的仿生学模型，通过多层结构可以学习到复杂的数据模式。具体到二元分类任务上，具体的网络结构如下所述：

- **输入层**：输入层接收特征数据，对于每个样本来说是多维向量，输入层的节点数等于特征数。
- **隐藏层**：隐藏层可以有一个或多个，每个层由若干神经元构成，连接前后层神经元的权重参数将在训练中被调整。常用 ReLU 激活函数。
- **输出层**：输出层是密集层，即普通的全连接层，将前一层的输出降维至两维，一维对应类别 1，另一维对应类别 0。
- **Softmax 层**：通过 Softmax 函数将输出层的值映射为概率分布，即两个节点的输出分别为属于类 0 和类 1 的概率。

在训练过程中，神经网络的参数通过前向传播计算输出，再通过交叉熵损失函数

计算损失，通过后向传播（基于梯度下降方法）优化参数。这种分类器对于高维特征、非线性数据模式、需要高预测准确性的应用场景十分契合，其强大的特征表达能力可以使系统的分类器对复杂的非线性数据做到高精度预测分类。

### 1.4.1. 密集层实现文本特征向量降维

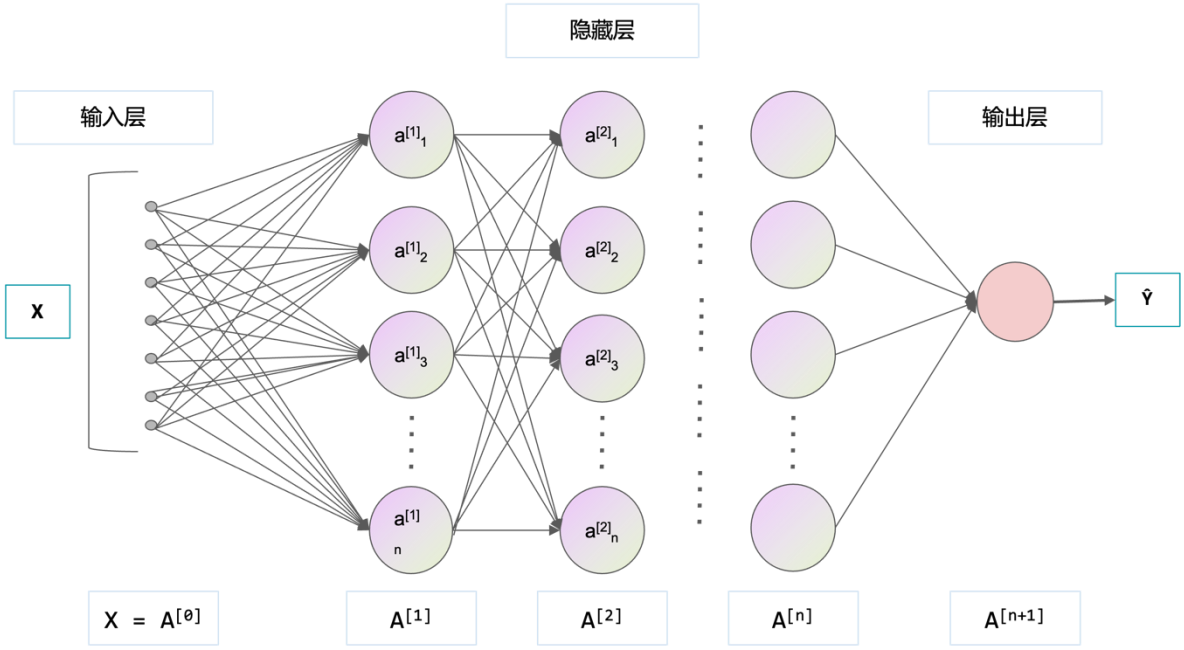


图 5 密集层结构图

为了减小高维特征向量的维度且尽可能保留其语义信息，密集层（Dense Layer），或称全连接层（Fully Connected Layer），被广泛应用于降维任务中。密集层是一种基础但极具效益的神经网络单元，通过线性变换和非线性激活函数，将输入高维特征向量映射到低维空间。以下将从理论、方法、实现细节等多个角度，深入探讨密集层在文本特征向量降维中的具体应用，其结构图如图 5 所示。

1. Bert 模型生成高维特征向量

Bert 模型，通过深度双向编码器结构，生成文本的高维特征表示。假设一段文本输入 Bert 模型后，输出的特征向量表示为 $H = \{h_1, h_2, \ldots h_n,\}$，其中每个 $h_i$ 表示词汇在上下文中的特征，且特征向量的维度通常为 768 或更高。对于该作品中的 NLP 任务，这样的高维特征向量既包含了语义信息，又带有丰富的上下文依赖，这无疑为下游任务提供了强大的信息支撑。然而，高维特征向量的缺点也十分明显，它增加了计算的复杂度和存储的压力，需要有效的降维方法来解决这个问题。

2. 密集层的结构与工作原理

密集层是一种全连接层，意味着它的每一层节点都与上一层节点全连接。具体地，假设输入向量为 x，权重矩阵为 W，偏置向量为 b，一个密集层的输出可以表示为 $y = f(Wx + b)$，其中 f 是非线性激活函数，如 ReLU（Rectified Linear Unit）、Sigmoid 或 Tanh。通过在全连接层中引入非线性激活函数，可以捕捉数据中的复杂模式和非线性关系。

3. 高维特征向量的降维机制

将 Bert 模型输出的高维特征向量输入密集层，进行线性变换和非线性激活，可以实现降维目的。假设原始特征向量维度为 768，通过设定密集层的输出维度，如 256，我们将高维特征映射到低维空间。这种维度压缩并非简单的丢弃信息，而是在尽可能保留原始数据的关键信息和特征的基础上，减少冗余和噪声。

密集层的训练通过反向传播算法（Backpropagation）进行，最小化损失函数（如均方误差或交叉熵损失）。在这一过程中，参数矩阵 W 和偏置向量 b 被不断调整，以实现最优的降维效果。密集层通过自适应的学习机制捕捉高维特征中的重要信息，并将其压缩到低维表示。

4. 参数初始化与正则化技术

密集层的权重矩阵和偏置向量需要初始化，以保证模型训练的有效性和稳定性。常用的初始化方法包括 Xavier 初始化和 He 初始化，通过设定初始参数的均值和方差，确保梯度在反向传播过程中稳定。一旦初始化完成，后续的训练过程便可以通过梯度下降等优化算法进行。

同时，为了防止模型过拟合（Overfitting），密集层在训练过程中常常引入正则化技术。L2 正则化通过在损失函数中加入权重矩阵的二范数，有效防止参数过大，进而提高模型的泛化能力。Dropout 技术则通过随机丢弃一部分神经元，强制模型更加鲁棒，不依赖于某些特殊神经元，提高模型的泛化性能。

5. 非线性激活函数的选择

在密集层中，选择合适的非线性激活函数是非常重要的，激活函数决定了模型能否学习到复杂的非线性关系。

综合考量 Tanh、Sigmoid、ReLU 三种激活函数的定义、优势和劣势，我们在该系统中最终选择 ReLU 作为密集层所用的激活函数，主要理由如下：

- **计算效率**：ReLU 计算简单，不涉及复杂的指数运算，计算效率极高，尤

其在大型神经网络中表现优异。

- **梯度传递**：ReLU 能有效缓解梯度消失问题，保证梯度在反向传播过程中的传递效率，使得深层神经网络的训练更加稳健和快速。
- **引入稀疏性**：ReLU 引入的稀疏性有助于减少参数更新，使得模型更为简洁和高效。
- **实际表现**：实验证明，ReLU 在大多数深度学习任务中（如图像分类、语音识别、自然语言处理等）效果优异，具有较强的泛化能力。

尽管 ReLU 存在“神经元死亡”问题，但这一问题相对可控，且通过使用一些变种激活函数（如 Leaky ReLU[51]、PReLU[52]等）可以进一步减小其影响。因此，我们选择 ReLU 作为本文中的激活函数是基于其在计算效率、梯度传递和实际效果上的综合优势。

在该系统中，Bert 模型输出的高维特征向量包含了丰富的上下文信息和语义特征。通过密集层的降维处理，这些信息被有效地压缩到低维空间，显著减少了计算开销和存储需求，同时在一定程度上强化了特征的判别能力和泛化性能，经过降维处理后模型的分类精度依然保持在较高水平。

### 1.4.2. 伪标签数据集生成

在获取到分类器对每段文本的分类结果后，我们需要将所有的预测结果进行伪标签生成，最终将所有生成的伪标签与对应的样本组合起来构成伪标签数据集。

具体来讲，在处理有限标注数据集的情况下，通过运用 MC Dropout 生成多个预测结果，我们可以为未标注样本生成伪标签，从而扩展训练数据集，提高模型的泛化性能和预测精度。

在实际操作中，我们通常面临着有标注数据有限而未标注数据充裕的情景。标注数据的获取往往成本高昂且费时费力，而未标注数据则相对易得。因此，如何有效利用未标注数据成为提升模型性能的关键之一。伪标签数据集生成技术正是利用未标注数据来扩展训练数据的一种有效方法。其核心思想在于利用一个已经过初步训练的模型为未标注数据进行预测，通过多次预测取多数票的方式生成伪标签，并将这些伪标签与对应的样本一同构成新的训练数据集。

具体来说，MC Dropout（Monte Carlo Dropout）是一种通过引入随机性来估计模型预测不确定性的方法。通常情况下，Dropout 在训练过程中随机丢弃一些神经元，

以防止过拟合。然而，在测试阶段，Dropout 通常被关闭，使得所有的神经元都参与到预测中。而在 MC Dropout 技术中，我们在测试阶段继续保持 Dropout 的开启，使得每次预测都带有随机性，从而生成多个不同的预测结果。这种方法能够有效地捕捉模型预测的不确定性，为生成伪标签提供了可靠的基础。针对每一个未标注样本，经过初步训练的神经网络模型利用 MC Dropout 进行多次预测。每次预测由于 Dropout 的随机性，结果可能会有所不同。假设我们对每个未标注样本进行 T 次预测，得到 T 个预测结果。随后，为了生成该样本的伪标签，我们采用多数投票法，即选择 T 个预测结果中出现次数最多的结果作为该样本的伪标签。这种方法利用了集成学习的思想，通过多次投票来提高伪标签的可靠性和准确性。具体而言，我们在随机 Dropout 下使用模型参数 $W_n$ 对基础模型执行 T 次随机前向传播。然后，对于未标记数据集 $D_u$ 中的每个文本 x ，我们获得一组伪标签 $\{\tilde{y}^1, \tilde{y}^2, \ldots, \tilde{y}^T\}$ 并计算多数预测作为最终的伪标签 $\tilde{y}$ ：

$$\begin{aligned} &\tilde{y}^t = arg\max\left[softmax(\mathcal{B}^{W_n^t}(x))\right] = c, \\ &\tilde{y} = \mathbb{I}(\sum_{t=1}^{T} \tilde{y}^t \geq \left\lfloor \frac{T}{|C|} \right\rfloor) = c, \\ &c \in C, 1 \leq t \leq T, \end{aligned} \tag{13}$$

其中 $W_n^t$ 是模型在第 n 个自训练过程中执行第 t 个 Dropout 操作时的参数，C 是不同样本类的预定义标签，$II$ 是指示函数。

在生成伪标签的过程中，预测次数 T 的选择。一般而言不宜过小，否则可能无法有效平滑模型预测的不确定性，同时也不应过大，否则计算开销会显著增加。在实践中，T 值通常选择在 10 到 50 之间。而对于模型在执行 MC Dropout 时的参数而言，Dropout 的概率，即每次丢弃神经元的比例是需要精确设置的。通常，Dropout 的概率在训练过程中已经设定好，但也可以针对生成伪标签的需求进行适当调整。较高的 Dropout 概率可能会增加模型预测的不确定性，从而产生更为多样化的预测结果；较低的 Dropout 概率则预测结果较为集中。

在得到所有未标注样本的伪标签后，我们将这些样本连同伪标签一起与原始标注数据进行合并，形成扩展后的训练数据集。伪标签样本的质量参差不齐，不能一概而论地将其等同于真实标注样本。因此，我们采用一些后处理技术应用于伪标签数据集，以提高其质量和可信度，包括根据模型对每个样本的预测置信度进行筛选，剔除置信

度过低的伪标签样本等。这样做能够有效减少错误伪标签对模型训练的不良影响。

通过伪标签生成过程，未标注数据被有效地利用起来，扩展了训练数据集，提高了模型的泛化能力。这一过程充分运用了 MC Dropout 的随机性和投票机制的稳健性，经过多次预测和投票生成可靠的伪标签，结合后续的置信度筛选和权重平衡等处理步骤，形成高质量的伪标签数据集，为模型训练提供了坚实的数据基础。在半监督学习和有限标注数据场景下，伪标签数据集生成无疑是一种行之有效的方法，极大地拓展了机器学习应用的边界和深度。

### 1.4.3. 恶意隐写载体预测

在系统通过深度学习模型对大量数据进行了分析和标记，通过多种方法如特征工程、监督学习和伪标签技术提升模型的预测准确性后，在这一步骤中，我们的神经网络模型输出了一批预测结果，这些结果表明各个数据样本是否包含潜在的恶意隐写内容。当模型给出的预测结果为 1 时，意味该样本极有可能是恶意隐写载体，此时我们需要进一步采取措施来警示和处理这种情况。

在这个过程中，我们的系统先由深度学习模型对输入数据进行逐个预测，产生一系列输出。其中，0 表示正常数据，1 表示检测到潜在的恶意隐写载体。模型的每一次预测结果都需要被详尽记录，确保后续处理能够准确跟进和执行。为了确保预测的可靠性，模型在每一次预测后都会输出一个置信度值，这个值反映了模型对当前预测结果的信心程度。高置信度值意味着结果的可靠性较高，反之则意味着需要进一步审查或人工干预。一旦模型产生了预测结果，我们进入检测和报警的关键环节。首先，我们需要实时监控所有预测结果，特别是那些结果为 1 的数据样本。在实际应用中，可以通过一个监控系统来持续跟踪和记录这些预测结果。一旦检测到某个样本的预测结果为 1，该监控系统便立即进行一系列的后续操作。

为了执行这一复杂的监控、报警和应急处理过程，我们借助现代化的机器学习平台和工具链 TensorFlow 以及各种开源的安全检测库和框架。这些工具集成了丰富的功能模块和强大的计算能力，为我们提供了灵活且高效的开发和实现环境。同时，通过搭建定制化的监控和报警系统，我们能够实现对恶意隐写载体的实时高效检测，大幅提升系统的安全性和可靠性。

## 1.5. 局部采样器及模型学习方法设计

在机器学习和数据挖掘领域，局部采样器及模型学习方法的设计是提升模型性能

的一种有效策略。特别是在处理大量数据集和复杂任务的情况下，合理设计采样器和优化模型学习方法，能够显著提高模型的训练效率和预测能力。在此我们详细探讨如何通过局部采样器计算每个样本对模型的信息增益，并基于这些信息定制采样权重，最后结合预测方差给每个样本设定损失权重，完成整体模型的优化学习。

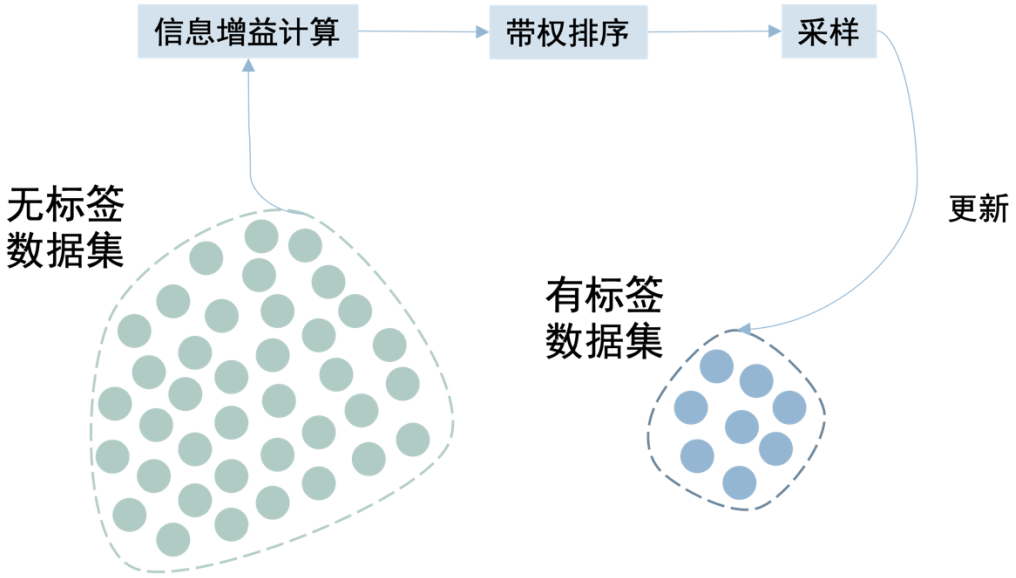


图 6 局部采样器流程图

在这之前，伪标签先已通过 MC Dropout 技术生成，具备了一定的置信度，但我们需要更进一步评估每个样本对模型的贡献。这一步骤的重要性在于，它能帮助我们识别出哪些样本对模型训练最具价值，即哪个样本能提升模型的泛化性能。我们通过计算每个样本对模型的信息增益来实现。信息增益是一种度量特征选择和数据节点分裂常用的指标，能够量化某个样本在特定分布下对模型预测准确性所带来的提升。对于我们生成的伪标签数据集中的每个样本，我们需要基于当前模型计算该样本的信息增益。信息增益的计算通常涉及模型前后的不确定性变化，可通过样本在模型预测分布中的体现来反映。简而言之，信息增益度量的是在引入特定样本之后，模型的不确定性减少了多少，通过这种度量，我们能够识别出那些对模型来说“信息含量”最高的样本，从而针对这些样本进行重点处理。

在得到了每个样本的信息增益后，我们下一步需要计算这些样本的排序权重。排序权重将帮助我们根据信息增益对样本进行排序，以便在后续的采样过程中优先选择具有较高信息增益的样本，我们对每个样本的信息增益值进行标准化，生成一个概率分布，然后基于这个概率分布对样本进行排序，分配对应的权重。排序权重不仅能指导采样过程中的优先次序，也为我们接下来的模型学习方法提供了准确的数据选择。

局部采样器则根据这些排序权重进行样本的选择。在这一过程中，采样器会依照权重分布，倾向于选择信息增益较高的样本进行学习。这一策略的优势在于，它能在有限的计算资源和时间内，最大化提升模型的训练效果。通过优先选择高价值样本，训练过程变得更加高效，并且有助于抑制过拟合现象，因为大多数信息含量低或者冗余的样本被适当地排除在外。在采样完之后，我们便形成了具体的需要用于微调我们模型的伪标签数据集，此时我们需要考虑模型的具体学习过程。为了进一步提升模型的学习效果，我们引入预测方差来设定每个样本的 loss 权重。预测方差反映了模型对特定样本预测结果的不确定性。较高的预测方差表示模型在对该样本进行预测时具有较高的不确定性，可能在训练过程中需要更多关注。因此，我们可以将预测方差作为样本的 loss 权重，高方差样本赋予更高的 loss 权重，引导模型在训练中更多关注这些不确定预测的样本。

在设定 loss 权重后，我们设计损失函数。本系统采用二元交叉熵损失函数，这是处理二分类问题时非常有效的损失度量方法。交叉熵在衡量预测概率分布与真实分布之间的差异上具有良好效果，能有效指导模型权重调整。具体的损失函数将结合前面设定的 loss 权重，使得高不确定性的样本对总损失贡献更大，从而在反向传播过程中，模型对这些样本进行更显著的权重调整。

### 1.5.1. 贝叶斯不一致主动学习[53]计算信息增益

贝叶斯不一致主动学习是针对数据样本选择的一种有效策略，通过不断优化样本选择过程来提升模型的学习效果。该方法的核心是计算信息增益，选择那些能够最大化信息增益的样本进行下一步训练。接下来我们将其分解为多个步骤进行描述，包括贝叶斯模型的构建、样本预测的概率分布、计算信息增益、选择高信息增益样本以及模型的迭代更新。该方法能够确保主动学习在数据利用效率和模型泛化性能上达到最佳效果。

信息增益是衡量某个样本对模型带来多少有用信息的指标。具体来说，信息增益计算样本在引入模型前后的不确定性变化。使用贝叶斯不一致主动学习为样本进行信息增益计算，需要两个关键步骤：

1. 计算样本当前的信息熵

信息熵反映了当前模型对样本预测的不确定性，可以通过样本的预测分布得到。例如，高斯过程的预测分布的方差可以直接用作不确定性量度，而贝叶斯神经网络可

通过多次采样方差估计不确定性。

2. 模拟该样本被标注和引入模型后的效果，计算新的信息熵

新的信息熵反映了模型在引入样本后的不确定性。我们通过贝叶斯更新公式，将新样本的预测分布与现有模型参数结合，得到更新后的模型后验分布，从而计算新的信息熵。

通过上述两个步骤，我们可以将信息增益表示为样本在引入前后信息熵的差异，公式表达为：信息增益 = 初始信息熵 - 更新后信息熵。这一差值代表了样本的“信息价值”。具体的计算公式如下所示：

$$\begin{aligned}&\mathrm{E}(\tilde{y}, W_n^T \mid x)\\&= -\sum_c (\frac{1}{T}\sum_t \hat{p}_c^t \log(\sum_t \hat{p}_c^t)) + \frac{1}{T}\sum_{t,c}(\hat{p}_c^t \log(\hat{p}_c^t)), c \in C,\end{aligned} \tag{14}$$

其中，$\hat{p}_c^t = p(\tilde{y}_i = c \mid B^{W^t}(x))$。

### 1.5.2. 样本排序权重计算

样本排序权重计算是根据某些指标计算出采样过程中每个样本自身的排序权重的过程。正确的样本排序权重计算方法不仅能确保每个样本对模型贡献最大化，还能有效提升模型的训练效率和性能。具体可分为：信息增益计算、排序权重的设定以及权重标准化策略三个步骤。

1. 得到各样本的信息增益

每个样本的信息增益为后续计算排序权重的基础。信息增益能量化某个样本对模型预测性能的提升程度，其中涉及到不确定性的度量。通过上述信息增益计算方法得出即可。

2. 样本排序权重设定

在计算了每个样本的信息增益后，我们需要设定样本排序权重。排序权重决定了在后续采样阶段，样本的优先级和选择概率。此过程的复杂性很大程度上取决于权重设定的合理性和计算方法的科学性，具体方法遵循下述步骤。

- **标准化处理信息增益**：首先，会对计算出的信息增益进行标准化处理，使其值落在一个合适的范围内。标准化处理确保不同样本的信息增益具备可比性。
- **设定基础权重**：以标准化的信息增益为基础，设定每个样本的初始排序权

重。这一步中可以通过线性或非线性映射方法，将信息增益转换为排序权重。在某些情况下，可选择凸函数或凹函数进行权重映射，以强调高信息增益或平衡中等信息增益的样本。

3. 权重标准化

权重标准化是一个必要步骤，确保设定的排序权重可以实际应用到采样过程中。标准化处理通过一系列数学转换，使得所有样本的权重总和为 1，便于后续概率性采样的实现。

- **初步标准化**：在该系统中，我们将 1 与将每个样本的原始权重作差值，之后求出所有差值的和，再使用各权重除以差值和，确保标准化后权重更加具有排序意义。
- **边界检查**：在标准化后需进行边界检查，避免因浮点误差或过度平滑导致某些权重值超出预期范围。

最终，我们系统中采用的权重计算公式如下所示：

$$r_i = \frac{1 - \mathrm{E}(\tilde{y}_i, W_n^T \mid x_i)}{\sum_{j=1}^{R} 1 - \mathrm{E}(\tilde{y}_j, W_n^T \mid x_j)}, \tag{15}$$

其中 R 表示 $D_u$ 中未标记样本的数量。$r_i$ 表示第 i 个样本的排序权重，值越小，意味着样本越难识别。然后我们可以从 $D_u$ 中选择样本，用它们的伪标签形成数据集$D_p$。

### 1.5.3. 二元交叉熵损失函数

在该系统中，我们采用带权重的二元交叉熵（Binary Cross-Entropy, BCE）损失函数，该损失函数在处理非均衡数据和复杂任务时非常有效。相较于传统的二元交叉熵损失函数，带权重的版本能针对不同样本赋予不同的损失权重，从而平衡训练过程中的样本贡献，特别是在不平衡数据集的分类任务中，即显示出了其独特的优势，通过为不同样本赋予不同的重要性，优化了模型的学习过程与分类性能。

二元交叉熵损失函数是处理二分类问题时常用的损失函数之一。其核心思想是通过度量预测分布与真实标签分布之间的差异来衡量模型的性能。具体而言，二元交叉熵计算的是模型输出的概率分布与实际标签分布的对数似然差异，目的是最小化这一差异，使得模型输出概率与实际类别标签更为接近。交叉熵损失对于模型输出的每一个类别概率进行独立的对数运算，通过加权平均的方法得出最终的损失值。在公式表

述上，普通的二元交叉熵损失涉及到两个部分：一个是正类样本对应的损失，另一个是负类样本对应的损失。对每一个样本，其损失是由模型预测的概率和实际标签之间的差异决定的，具体为：如果样本属于正类损失为负对数概率，如果样本属于负类损失为负对数反概率。因此，二元交叉熵非常强调预测概率的准确性。计算公式如下所示：

$$Loss' = -\frac{1}{N_s'}\sum_{i=1}^{N_s'} w_i \left[\tilde{y}_i \cdot \log(\hat{y}_i) + (1-\tilde{y}_i)\cdot \log(1-\hat{y}_i)\right], \tag{16}$$

其中 $N_s'$ 是批量大小，$\hat{y}_i$ 表示预测标签。

但考虑到普通的二元交叉熵损失函数可能会导致模型倾向于优先预测占比高的类别，因为这样可以使总体损失最小化，但却无助于少数类别的准确分类。此时，我们在系统中引入权重机制。带权重的二元交叉熵损失函数旨在解决上述问题，通过为不同类别的样本设定不同的重要性权重，增强模型对不平衡数据的敏感性。权重的设置可以基于样本数量、类别的重要性以及其他先验知识。例如，对于正类样本少于负类样本的情况，我们可以为正类样本设定更高的权重，以增强其在损失计算中的影响力。这样，模型在训练过程中会更多关注这些权重较大的样本，从而提升分类性能。

具体地，带权重的二元交叉熵损失函数在计算时，将每个样本的损失乘以其对应的权重，然后再进行加权平均。通过这种方式，不同类别样本的损失能在总损失中占据相应的比例，确保模型在优化过程中能更均衡地考虑各类别样本的影响。在该系统中，我们采用计算样本预测方差的方法来获取每个样本的 loss 权重，具体的计算公式如下所示：

$$w_i = \alpha \mathrm{Var}(\tilde{y}_i) = \alpha \frac{\sum_{t=1}^{T}(B^{W_n^t}(x_i) - \overline{B^{W_n^t}(x_i)})^2}{T}, \tag{17}$$

其中 $\alpha$ 是调整权重总体大小的系数。

对于带权重的 loss 函数计算方式，我们不难看出虽然全过程复杂，但通过科学的权重设定策略、优化和计算技巧，可有效克服数据不平衡和复杂性带来的挑战，确保模型在实际应用中的鲁棒性和泛化能力。

### 1.5.4. 半监督模型微调技术

半监督学习技术能够利用大量无标签数据和少量有标签数据来提升模型的训练效果。自训练（self-training）则是一种常见的半监督学习策略，这种方法主要通过逐

步生成伪标签来逐步微调基础模型。在我们的系统中，我们设计了以下几个步骤来完成模型的半监督微调。

1. 基础模型微调

在自训练方法中，整个过程的首要步骤是利用有标签的数据集对基础模型进行初始微调。这一阶段的目标在于使模型具备初步的分类能力，从而为后续的伪标签生成和自训练打下基础。数据准备和预处理是关键步骤，需要对有标签数据进行严格的清洗、特征提取和标准化，以确保输入数据的高质量。我们选择 Bert 与分类器的组合模型作为我们的基础模型。在设定优化策略时，损失函数通常选择交叉熵损失，优化算法则选用 Adam 以优化模型参数。训练过程中，将有标签数据划分为训练集和验证集，通过交叉验证和早停策略，确保模型能有效地微调且避免过拟合，从而达到初步的理想性能。

2. 伪标签数据集的构造

完成初始模型微调后，利用基础模型对无标签数据集进行预测，并生成伪标签。具体而言，通过模型对无标签样本的预测概率，选择预测置信度最高的类别作为伪标签。在这一步中，我们从三个主流社交平台（分别是：Tweet、IMDB、News）中构造无标签数据，确保了数据的多样性和与任务相关的覆盖广泛度。通过预测和置信度评估效筛选出高置信度的样本，并将其伪标签添加到原有的有标签数据集中。通过这一步骤，形成一个扩展的训练集，使得模型能够同时学习真实标签和伪标签数据，为进一步微调模型奠定基础。

3. 样本选择与二次训练

新生成的扩展训练集包含了原有的真实标签样本和新生成的伪标签样本。为了对模型进行进一步的微调，我们选择了合适的样本选择策略和训练方法。为增强高置信度样本的影响，我们在模型中的损失函数运行过程中引入样本权重的计算，将不同置信度的样本赋予不同的重要性权重，从而优化模型参数。最后结合迭代优化和交叉验证策略，这一步骤确保模型在扩展训练集（伪标签数据集）上得到进一步的优化。

4. 模型收敛与效果评估

随着迭代训练的进行，模型性能逐步提升并趋于收敛。在这个过程中，需要监控模型的训练效果与收敛情况。我们选择准确率与 F1 得分作为我们系统模型的性能评估指标，全面评估模型在训练集和验证集上的表现。若模型在验证集上的性能不再显

著提升，表明其可能已趋于收敛，为了避免过拟合问题，我们采用带有耐心值的早停策略，我们设定在验证集性能持平时一定轮数内及时停止训练。最终，我们通过保存每次迭代的最佳模型，最终选择性能最优的模型作为最终结果。

自训练方法在半监督学习模型微调中的应用，能够充分利用大量无标签数据和有限的有标签数据，通过迭代更新和伪标签生成，实现模型性能的显著提升。上面四个步骤在整个过程中均起到了至关重要的作用。在实际系统设计过程中，我们通过合理设计和优化每个环节使得半监督模型微调技术可以有效解决数据不平衡和标签稀缺等问题，为整个恶意隐写载体检测系统提供高效可靠的模型训练方案。