	Repo	Defect ID	Defect Name/Link	Defect Description	Scenario analysis	ODC Type	HEM	Confidence
1	Keras	#18274	fix class_names arg checking thinko #18724	The code is supposed to pass a ValueError if the dataset class names are not a subset of the subdirectory list. However, the code mistakenly passes a value error if the dataset class names ARE a subset of the subdirectory list.	The current task requires Rule X (passing a ValueError if the dataset class names are not a subset of the subdirectory list). The person uses Rule Y (passing a ValueError if the dataset class names are a subset of the subdirectory list) instead of Rule X.	Assignment	HEM2	High
2	Keras	#18829	Made masked losses compatible with masked nans #18829	A function that masks values is supposed to set values to zero, even if they are NaN (undefined). The code incorrectly multiplies masked values by zero, (leaving NaNs as Nans) instead of setting them to zero.	The current task requires Rule X (set NaN values to zero). The person uses Rule Y (leaves NaN values as NaNs) instead of Rule X.	Algorithm	HEM2	Medium
3	Keras	#18734	Pass dtype to cumsum and cumprod #18734	keras.ops.cumsum and keras.ops.cumprod both call numpy backend without specifying the dtype. The backend converts all ints to int64s when the dtype is not specified, leading to inconsistency errors.	The competing rule (not needing to specify or standardize the dtype) is very "strong" in that it is used successfully very often	Interface	HEM3	High
4	Keras	#18766	Set noise_shape to None in attention.py #18766	keras.layers.Attention has the line noise_shape=self.noise_shape that fails since Attention has no attribute noise_shape.	The competing rule is in a "First Exception": the novel circumstance is that the layer doesn't have a noise_shape attribute.	Assignment?	HEM3	Medium
5	Keras	#19093	Fix for _calculate_score_mask in Attention Layer #19093	The function attempts to perform operations with a value, v_mask, that can be set to None. However, the function doesn't check whether the value is None before attempting the operation, thus resulting in a ValueError when v_mask is None.	The function contains multiple features, including an input value that can be None, but the person omits a check for None values, which has less salience than the other features of the function	Checking	HEM5	Medium
6	Keras	#19088	Correct keras.ops.take behaviour with axis=None #19088	If Axis is None, the function is supposed to flatten the data and then set the axis to 0. The current code doesn't flatten the data, resulting in incorrect indexing.	The current task requires Rule X (flatten the data). The person uses Rule Y (not flattening the data) instead of Rule X.	Algorithm	HEM2	Medium
7	Keras	#19057	raise exception on unknown padding for conv #19057	The only acceptable values for Padding are 'valid' or 'same', but the operation util doesn't throw an exception for unknown types of padding. This leads to errors in conv layers that depend on this util.	The function requires a test on padding values, which the developer missed.	Checking	HEM6	Medium
8	Keras	#19102	Fix wrong gradients for torch batch_norm and moments #19102	The code is supposed to use gradients from the mean to help compute the batch normalization. However, the operation mistakenly cuts off gradients from the mean in the batch_normalization function, so they are not used.	The current task requires Rule X (keep gradients flowing). The person uses Rule Y (cut off gradients) instead of Rule X.	Algorithm	HEM2	High
9	Keras	#18975	Fix compiled crossentropy #18975	When setting loss to CrossEntropy, there is a missing edge case for when the output is a 1-element list.	The task requires the developer to review their work, which should have 4 edge cases if there is only 1 output, but they only implemented 3 of the edge cases.	Checking	HEM6	High
10	Keras	#18986	Fix clipnorm for TF models with Embedding layer(s). #18986	The code uses the * operator for multiplication, which causes a TypeError when the input is tf.IndexedSlices. For this new situation, the code needs to use ops.multiply().	The task requires the ops.multiply() function for multiplication, the developer* instead because * has similar functionality to ops.multiply()	Algorithm	HEM9	Low
11	Keras	#19217	Fix test_case.run_layer which was not training with training=True. #19217	TestModel.call() is missing the training argument, so training currently isn't passed to the layer during the training test.	Current task contains multiple features, but one of the function arguments was missed. The developer focused on other (more salient) features of the function.	Assignment	HEM5	Low
12	Keras	#19183	Keras 3 SpectralNormalization() #19183	SpectralNormalization had an if that should have been a ops.cond	The competing rule (using an "if" statement) is very "strong" in that it is used successfully very often, and the "First Exception" (using "ops.cond" instead) is a new feature	Algorithm	HEM1	Low
13	Keras	#18835	Bug fix for TF backend using random zoom with flatten #18835	The RandomZoom layer doesn't maintain the original tensor shape, which causes an error when followed by a Flatten layer which requires the original tensor shape.	The developer forgot to ensure that the tensor shape is maintained after it is passed through multiple different features, so attention was not given to the right feature.	Checking	HEM5	Medium
14	Keras	#18828	Fixed layer.losses such that it recurses into sublayer losses #18828	The function is supposed to be recursive. However, it currently only recurses 1-child deep, so nested sublayers are mistakenly ignored and it does not have a recursive implementation.	The developer focused on the wrong features: the function mistakenly only looks at the first layer of child layers, not the sublayers.	Algorithm	HEM5	Medium
15	Keras	#18825	torch/numpy.py: Fix bool sort on CUDA #18825	On CUDA devices, bool values need to be converted to uint8 because the sort function does not support bool dtype on CUDA.	The developer mistakenly didn't convert booleans to uint8s because they didn't realize the sort function doesn't support this dtype on CUDA.	Assignment	Unknown	High

16	Keras	#18822	EarlyStopping: Restore best weights after the last epoch #18822	The EarlyStopping callback mistakenly only restores the best weights if it requests the stopping itself. But the callback should restore the best weights whenever training stops, regardless of the reason why it stopped.	The current task requires Rule X (restore best weights if it requests stopping itself). The person uses Rule Y (restore best weights no matter why training stopped) instead of Rule X.	Algorithm	HEM2	Medium
17	Keras	#18795	Fix numerical issues in TF and JAX moments implementation. #18795	This is a numerical issue where the subtraction operation does not guarantee a non-negative result given float precision, so it is supposed to be clamped to 0.	The developer mistakenly didn't clamp the subtraction operation to 0 as they didn't realize the subtraction operation doesn't guarantee a non-negative result given float precision	Algorithm	Unknown	Low
18	Keras	#18775	Bug fix for the JAX ops.arange warning #18775	The dtype is not standardized in the arange function, leading to errors when dtype is "int" because it is not converted correctly.	The developer paid attention to other features, omitting the standardization of the dtype	Algorithm	HEM5	Low
19	Keras	#18773	Fix torch breakage for convert_to_tensor([]) #18773	If no dtypes are provided, the result_type function is supposed to default to float because that is the expected behavior of convert_to_tensor function.	The current task requires Rule X (default to float if no dtype is provided). The person uses Rule Y (no default dtype) instead of Rule X.	Algorithm	HEM2	High
20	Keras	#18641	Fix torch reshape #18641	The reshape function takes in an array and a new shape. The function fails if the shape is not a list or tuple, so it should be set to list before being reshaped.	The developer failed to take a test condition into consideration: that the input might not be a list or tuple.	Checking	HEM6	High
21	Keras	#18629	Switch from tensor reshape to ops.reshape #18628	The code mistakenly calls x.reshape(shape) when x doesn't always have a reshape function. Instead, the code needs to call backend.numpy.reshape(x, new_shape).	The competing rule (calling the reshape() function) is "Strong": it has been used successfully often in the past, but this task requires calling backend.numpy.reshape() instead	Interface	HEM3	Medium
22	Keras	#18603	Fixed clone_model with no inputs #18603	When calling the function with no inputs, the function mistakenly creates inputs with an additional leading dimension.	The developer paid attention to other features in this task, not paying attention to the accidental leading dimension of the input, which is less psychologically salient than other parts of the function	Assignment	HEM5	High
23	Keras	#18123	https://github.com/keras-team/keras/pull/18123	The model should output the same dtype as the last layer, but it outputs as float16.	The current task requires Rule X (output same dtype as previous layer). The person uses Rule Y (outputs float16 dtype) instead of Rule X.	Interface	HEM2	Low
24	Keras	#18193	add MultiStepLR; correct CosineDecay according to the original paper and pytorch by Edenzzzz · Pull Request #18193 · keras-team/keras (github.com)	The implementation of CosineDecay mistakenly calculates the decay batch-wise, when it is supposed to be calculated epoch-wise.	The current task requires Rule X (calculate decay epoch-wise). The person uses Rule Y (calculate decay batch-wise) instead of Rule X.	Algorithm	HEM2	Medium
25	Keras	#18737	fixes TypeError by colbybanbury · Pull Request #18737 · keras-team/keras (github.com)	The copy() function is called with an 'overwrite' argument that is not an argument of this function.	The developer lacks knowledge relevant to the current task (shown by calling an argument that doesn't exist)	Assignment	HEM3	Medium
26	Keras	#18675	Fix numpy conversion for tf MirroredVariable. by qlzh727 · Pull Request #18675 · keras-team/keras (github.com)	Numpy requires this method to return an array, but in the case of scalar values, it will fail the type checking, so they need to be converted to a 0d array.	The developer failed to take a test condition into consideration: scalar values, which are a corner case and cause the array to fail the type check	Assignment	HEM6	Medium
27	Keras	#18585	Fix torch gpu tensors in `tf.data.Dataset` - `flatten_test.py` by sampathweb · Pull Request #18585 · keras-team/keras (github.com)	When a PyTorch GPU tensor is passed to the function, Tensorflow tries to convert it to a Tensor, but the tensor needs to be brought on the CPU before it can be converted.	The competing rule (converting torch tensors to tensorflow tensors) is "Strong" (used successfully in the past very often) but in this case, since the torch tensor is from the GPU, it needs to be brought to the CPU first (new exception)	Function	HEM1	Medium
28	Keras	#19118	Fix bug in broadcast_to Op with Jax backend when keras variable as input by SuryanarayanaY · Pull Request #19118 · keras-team/keras (github.com)	The function expects Array as input and fails with Tensor input. So the code is supposed to explicitly convert the input to Array before passing it into the function.	The developer failed to take into consideration that the input could be a Tensor instead of an array, so this is a missed input validation edge case	Assignment	HEM6	Medium
29	Keras	#18550	Fix for torch2.1.0 by james77777778 · Pull Request #18550 · keras-team/keras (github.com)	The resize function takes in a list for the new shape and fails when the input is nd.arrays. The code needs to explicitly call the tolist() function to make the shape a list.	The developer didn't pay attention to the right features of the task, resulting in them omitting the explicit tolist() function call	Assignment	HEM5	Low

30	Keras	#18550	Fix for torch2.1.0 by james77777778 · Pull Request #18550 · keras-team/keras (github.com)	In the optimizer, a variable that requires gradients is being used in an in-place operation. @torch.no_grad needs to be called before the optimizer is called.	The developer didn't pay attention to the right features of the task, resulting in them omitting the "@torch.no_grad" call	Algorithm	HEM5	High
31	Keras	#18547	Fix issues with tensordot on torch backend by mattdangerw · Pull Request #18547 · keras-team/keras (github.com)	Torch only handles dimensions of tuples but numpy accepts dimensions of a single tuple. The code need to check for if the dimensions are a list or tuple and ensures they are in the Torch format of tuples.	The developer failed to take a test condition (dimensions of a single tuple) into consideration	Checking	HEM6	Medium
32	Kymatio	#1022	MAINT use lr_scheduler instead of resetting optimizer #1022	The code is supposed to update the optimizer learning rate every 20 epochs but instead it redefines the optimizer every 20 epochs (erasing data from momentum variables)	The developer didn't pay attention to the right features, thus accidentally redefining the optimizer inside the for-loop instead of updating the optimizer	Assignment	HEM5	Low
33	Pytorch	#110408	fix batch_isend_irecv example incorrect usage #110408	Mismatched dtypes silently lead to wrong outputs because the developer didn't specify dtype (which is common to not need) in the function calls.	The developer failed to take a test condition (dimensions of a single tuple) into consideration	Documentation	HEM6	Medium
34	Pytorch	#110454	fix(optim): adagrad sparse multitensor incorrect early exit #110454	The use of "return" instead of "continue" causes the function to incorrectly exit early when len(groups)>1. The developer commented, “We return here to prevent the function from continuing to run. We could have it run and just return nothing after, but it doesn't seem like that would make a big difference. I may be missing something though and you should let me know.”	The current task requires the developer to use the "continue" command, but this did not exist in their knowledge base.	Algorithm	HEM3	High
35	Pytorch	#110562	feat(optim): add SGD sparse multitensor to testing path #110562	The multitensor case is missing from the testing path.	The developer failed to take a test condition (multitensor edge case) into consideration	Algorithm	HEM6	High
36	Pytorch	#100811	Handle tail 0-size tensor appropriately in MultiTensorApply #100811	Passing an empty tensor in the optimizer prevents parameter updates.	The developer failed to take a test condition (empty tensors) into consideration	Algorithm	HEM6	Medium
37	Pytorch	#110420	[sparse] Fix semi-structured sparse shape mismatch bug #110420	The returned matrix is an incorrect size in the case of multi-dimensional input (>2d).	The developer failed to take a test condition (multidimensional input) into consideration	Algorithm	HEM6	Medium
38	Pytorch	#110850	Fix num_batches_tracked of BatchNorm when load_state_dict #110850	BatchNorm layer "num_batches_tracked" is overwritten with default value 0.	The developer paid attention to the wrong features and didn't notice that the num_batches_tracked feature is implemented incorrectly and being overwritten	Assignment	HEM5	Medium
39	Pytorch	#111205	Use 'device' argument in test_sparse.py::TestSparseAnyCUDA::test_as_sparse_gradcheck_* #111584	Device argument was missed, causing the test function to always run on the default device (cpu) if another default torch device was not previously configuresd.	The developer paid attention to the wrong features, missing the device argument, which is less psychologically salient than other features in the function	Function	HEM5	Medium
40	Pytorch	#111427	[fix] accounting for dilation in pool padding assertion #111427	Dilated max-pooling fails due to padding check, but it should work even when the amount of padding is greater than half the kernel size because dilation imitates larger kernels.	The current task requires Rule X (the limit padding size is 1 + (kernel_size) - 1 * dilation). The person uses Rule Y (the limit padding size is kernel_size * dilation) instead of Rule X.	Algorithm Checking	HEM2 HEM8	High