Design a Neural network with various optimization algorithms and analyse their performance using Keras.

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.optimizers import SGD, Adam, RMSprop

# Load and preprocess the data
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

# Build neural network function
def build_model(optimizer):
    model = Sequential([
        Flatten(input_shape=(28, 28)),
        Dense(128, activation='relu'),
        Dense(10, activation='softmax')
    ])
    model.compile(optimizer=optimizer,
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
    return model

# Define optimization algorithms
optimizers = {
    'SGD': SGD(),
    'Adam': Adam(),
    'RMSprop': RMSprop()
}

# Train and evaluate models
results = {}
for optimizer_name, optimizer in optimizers.items():
    print(f'Training model with {optimizer_name} optimizer...')
    model = build_model(optimizer)
    history = model.fit(x_train, y_train, epochs=5, validation_split=0.2, verbose=0)
    results[optimizer_name] = history.history

# Evaluate models on test data
print('\nEvaluation on test data:')
for optimizer_name, history in results.items():
    model = build_model(optimizers[optimizer_name])
    loss, accuracy = model.evaluate(x_test, y_test, verbose=0)
    print(f'{optimizer_name} optimizer - Test accuracy: {accuracy * 100:.2f}%')
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
Training model with SGD optimizer...
Training model with Adam optimizer...
Training model with RMSprop optimizer...

Evaluation on test data:
SGD optimizer - Test accuracy: 11.74%
Adam optimizer - Test accuracy: 12.03%
RMSprop optimizer - Test accuracy: 8.28%
```