that large organizations can reduce risks associated with operating critical business functions on legacy systems.

## SUMMARY

**[0006]** Aspects of the disclosure relate to methods, apparatuses, and/or systems for translating a first coding language into a second coding language.

**[0007]** In an aspect, the disclosure relates to a method for translating a first coding language into a second coding language, including: training, by a processor, a first machine learning (ML) model at least in part on a first coding language specific data set relating to the first coding language, in which the first ML model is trained to translate one or more code sets of the first coding language to respective one or more code sets of the second coding language; using the first ML model, generating, by the processor, at least one unit test case, in which the at least one unit test case runs the one or more code sets of the second coding language in parallel with the one or more code sets of the first coding language; iteratively testing and refining, by the processor, the first ML model based at least in part on a maturity level of the first ML model until a maturity threshold is reached; and upon reaching the maturity threshold, containerizing by the processor, the one or more code sets of the second coding language into an application.

**[0008]** In some examples, the techniques described herein relate to a method, in which the first coding language specific data set includes one or more of at least one of a language reference document, library, historical input file, historical output file, runtime log, parameter set, or control point, relating to a first coding language.

**[0009]** In some examples, the techniques described herein relate to a method, in which the first coding language is Common Business-Oriented Language (COBOL).

**[0010]** In some examples, the techniques described herein relate to a method, in which the second coding language is one of Java, Golang, Python, Angular, or C++.

**[0011]** In some examples, the techniques described herein relate to a method, in which the first machine learning model is a Natural Language Model (NLM).

**[0012]** In some examples, the techniques described herein relate to a method, in which iteratively testing the first ML model includes: implementing a plurality of iterative regression tests based on historical input data of at least one of the one or more code sets of the first coding language and comparing corresponding output data of the first ML model against historical output of the at least one of the one or more code sets of the first coding language.

**[0013]** In some examples, the techniques described herein relate to a method, in which iteratively refining the first ML model includes: executing, by the processor, one or more debugging techniques; and updating the first ML model based on the one or more executed debugging techniques.

**[0014]** In some examples, the techniques described herein relate to a method, further including: dynamically scaling, by the processor, one or more containerized applications based at least in part on one or more of a second ML model or at least one second unit test case that has reached the maturity threshold.

**[0015]** In some examples, the techniques described herein relate to a method, further including: tracking, by the processor, progress of the at least one test case based at least in part on the maturity level of the first ML model.

**[0016]** In an aspect, the disclosure relates to a system for translating a first coding language into a second coding language, including: a computer having a processor and a memory; and one or more code sets stored in the memory and executed by the processor, which, when executed, configure the processor to: train a first machine learning (ML) model at least in part on a first coding language specific data set relating to the first coding language, in which the first ML model is trained to translate one or more code sets of the first coding language to respective one or more code sets of the second coding language; using the first ML model, generate at least one unit test case, in which the at least one unit test case runs the one or more code sets of the second coding language in parallel with the one or more code sets of the first coding language; iteratively test and refine the first ML model based at least in part on a maturity level of the first ML model until a maturity threshold is reached; and upon reaching the maturity threshold, containerize the one or more code sets of the second coding language into an application.

**[0017]** In some examples, the techniques described herein relate to a system, in which the first coding language specific data set includes one or more of at least one of a language reference document, library, historical input file, historical output file, runtime log, parameter set, or control point, relating to a first coding language.

**[0018]** In some examples, the techniques described herein relate to a system, in which the first coding language is Common Business-Oriented Language (COBOL).

**[0019]** In some examples, the techniques described herein relate to a system, in which the second coding language is one of Java, Golang, Python, Angular, or C++.

**[0020]** In some examples, the techniques described herein relate to a system, in which the first machine learning model is a Natural Language Model (NLM).

**[0021]** In some examples, the techniques described herein relate to a system, in which, when iteratively testing the first ML model, the processor is further configured to: implement a plurality of iterative regression tests based on historical input data of at least one of the one or more code sets of the first coding language and comparing corresponding output data of the first ML model against historical output of the at least one of the one or more code sets of the first coding language.

**[0022]** In some examples, the techniques described herein relate to a system, in which when iteratively refining the first ML model, the processor is further configured to: execute one or more debugging techniques; and update the first ML model based on the one or more executed debugging techniques.

**[0023]** In some examples, the techniques described herein relate to a system, in which the processor is further configured to: dynamically scale one or more containerized applications based at least in part on one or more of a second ML model or at least one second unit test case that has reached the maturity threshold.

**[0024]** In some examples, the techniques described herein relate to a system, in which the processor is further configured to: track progress of the at least one test case based at least in part on the maturity level of the first ML model.

**[0025]** In an aspect, the disclosure relates to a non-transitory computer-readable medium storing computer-program instructions that, when executed by one or more processors, cause the one or more processors to effectuate operations including: training a first machine learning (ML) model at least in part on a first coding language specific data set relating to the first coding language, in which the first ML model is trained to translate one or more code sets of the first coding language to respective one or more code sets of the second coding language; using the first ML model, generating at least one unit test case, in which the at least one unit test case runs the one or more code sets of the second coding language in parallel with the one or more code sets of the first coding language; iteratively testing and refining the first ML model based at least in part on a maturity level of the first ML model until a maturity threshold is reached; and upon reaching the maturity threshold, containerizing the one or more code sets of the second coding language into an application.

**[0026]** In some examples, the techniques described herein relate to a non-transitory computer-readable medium, in which the first coding language is Common Business-Oriented Language (COBOL); and in which the second coding language is one of Java, Golang, Python, Angular, or C++.

**[0027]** Various other aspects, features, and advantages will be apparent through the detailed description and the drawings attached hereto. It is also to be understood that both the foregoing general description and the following detailed description are exemplary and not restrictive of the scope of the disclosure.

## BRIEF DESCRIPTION OF THE FIGURES

**[0028]** FIG. 1 depicts an illustrative system for translating a first coding language into a second coding language, in accordance with at least one embodiment;

**[0029]** FIG. 2 illustrates an example code translation application in accordance with at least one embodiment;

**[0030]** FIG. 3 depicts an example method for translating a first coding language into a second coding language, in accordance with at least one embodiment;