1. Обзор методов прогнозирования системных вызовов

1.1 Описание существующих методов:

1.1.1 Статистические методы:

1.1.1.1 Предиктор последнего значения

Предиктор последнего значения [1] предсказывает, что значение текущего экземпляра инструкции будет таким же, как значение, полученное предыдущим экземпляром. Этот предиктор применяет функцию идентификации к ранее наблюдаемому значению и использует его в качестве следующего прогноза. Этот метод довольно простой и полезен, когда в данных наблюдается устойчивая тенденция и незначительная изменчивость.

1.1.1.2 Предиктор на основе таблиц

Программы при выполнении демонстрируют различную степень периодичности, что указывает на то, что модели поведения повторяются с течением времени. Предикторы на основе таблиц [1] разработаны на использовании этой повторяемости. Они используют историю прошлого поведения в качестве индекса в таблице прогнозирования. Предсказание, сохраненное в таблице является значением, которое соответствовало истории поведения в прошлом.

При выполнении новой рабочей нагрузки предикторы проходят начальную фазу обучения, чтобы заполнить таблицу прогнозов историческими шаблонами. Для получения прогнозов на этапе обучения может использоваться схема Предиктор последнего значения. Предикторы адаптируются к изменениям в изученном поведении, обновляя записи в таблице, но необходимо соблюдать осторожность, чтобы избежать обновления таблицы ложными значениями из-за шумовых пиков. В дополнение к прогнозируемому значению, в каждой записи таблицы хранится последнее фактическое значение, которое наблюдалось в истории. После получения нового фактического значения, таблица обновляется путем установки на наиболее частое из трех значений: прогнозируемого и последнего значения, сохраненные в таблице, и нового фактического значения.

1.1.2 Марковские модели:

Основаны на предположении, что будущее состояние зависит только от текущего состояния. Могут учитывать ограниченный контекст.

1.1.2.1 Марковские цепи

1.1.2.1 Скрытые марковские модели (HMM)

1.1.3 Машинное обучение:

1.1.3.1 Предиктор на основе дерева решений

Предиктор на основе дерева решений используются методы машинного обучения (дерево решений) построенное при помощи алгоритма C4.5. Для построения датасета используется набор определенных статических и динамических атрибутов процессов во время их выполнения. В работе [3] были проведены исследования по выявлению наиболее важных статических и динамических характеристик процессов, которые могут наилучшим образом помочь в прогнозировании времени загрузки процессора и минимизировать время выполнения процесса. Результатом работы стал набор атрибутов для построения датасета, представленный в таблице 1, который был получен с помощью команд readelf и size.

Таблица 1 – Набор атрибутов для построения датасета для предиктора на основе дерева решений

Атрибут	Его ранг	Обозначение
InputSize	1	Значение размера входных данных, зависящее от типа программы
ProgramName	2	Название программы и номинальный атрибут
BSS	3	Информация о размере неинициализированных данных
RoData	4	Размер данных (байт), доступный только для чтения, который обычно используется для создания недоступного для записи сегмента в образе процесса
Text	5	Текст или исполняемые инструкции
InputType	6	Тип входного номинального значения

Обучение осуществляется путем анализа статических и динамических атрибутов процессов, представленных в таблице.

1.1.4 Нейронные сети:

1.2 Критерии сравнения:

- **Способность улавливать зависимости:** Насколько хорошо модель моделирует зависимости между системными вызовами в последовательности.
- **Масштабируемость:** Способность модели обрабатывать большие объемы данных и длинные последовательности.
- **Вычислительная сложность:** Ресурсы, необходимые для обучения и использования модели.

1.3 Таблица сравнения:

1.4 Выводы по итогам сравнений:

2. Сравнение нейронных сетей для прогнозирования системных вызовов

2.1 Описание аналогов

2.1.1 Предсказание на основе Sequence-to-Sequence модели

В своей архитектуре Sequence-to-Sequence модель [4] использует рекуррентные нейронные сети (RNN) для обработки последовательностей входных и выходных данных. В этой работе для улучшения производительности базовых классификаторов обнаружения вторжений авторы объединили предсказанную последовательность с вызванной последовательностью, чтобы сформировать расширенные входные данные. Как показано на рис. 1, предсказанная последовательность предоставляет классификаторам дополнительную информацию для улучшения возможности определения характера последовательности системных вызовов.

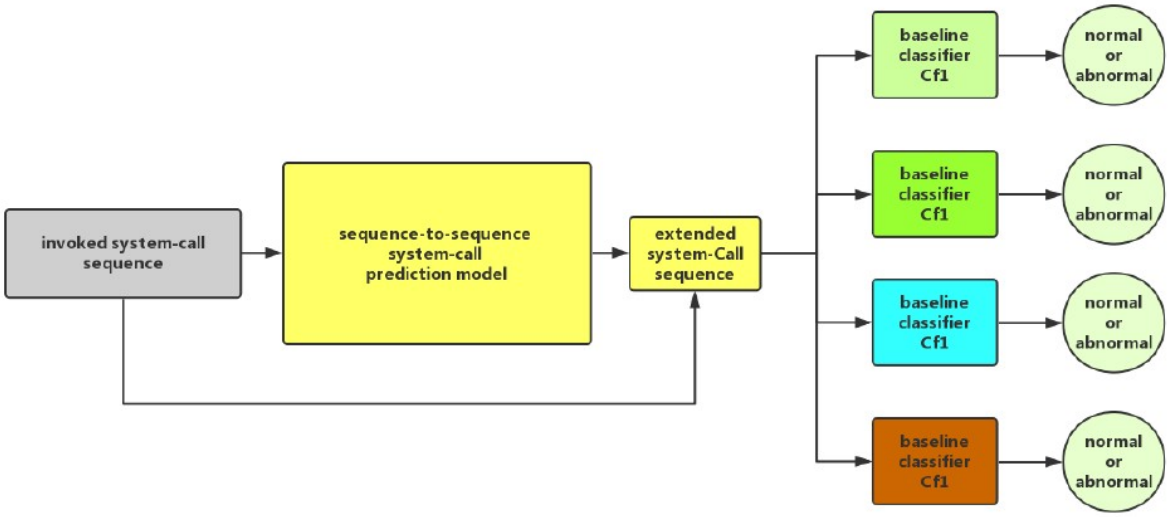

Рисунок 1 – Схема работы Sequence-to-Sequence модели

Авторы рассматривают трассировки системных вызовов, создаваемые во время работы программы, как набор языковых предложений, которые процесс использует для взаимодействия с операционной системой. Для генерации семантически обоснованных последовательностей системных вызовов внедряется структура sequence-to-sequence языковой модели RNN с «механизмом внимания» (attention) (кодер выполняет операцию прямого распространения). Механизм attention позволяет подчеркнуть влияние некоторых определенных системных вызовов, которые лучше отражают намерения программы, на конкретное декодирование системных вызовов. Таким образом данный метод использует информацию о входных данных, которые наиболее важны при декодировании каждого системного вызова.

Архитектура модели sequence-to-sequence с attention показана на рис. 2.

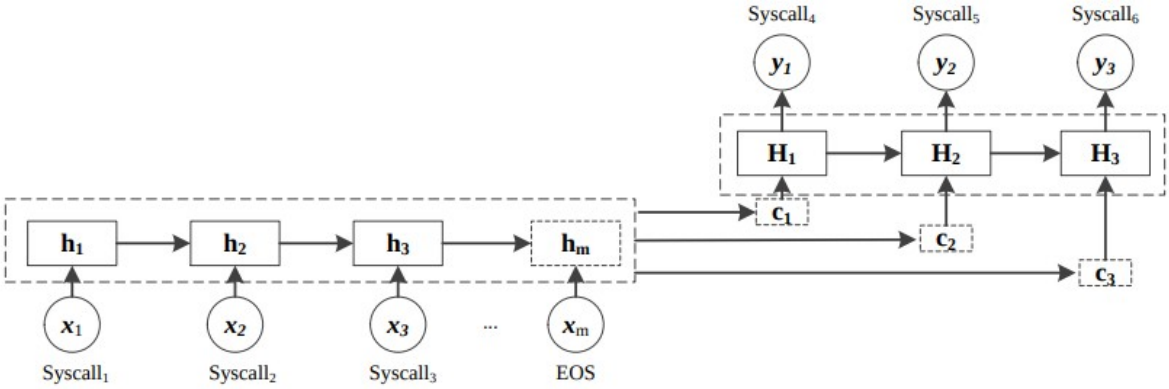

Рисунок 2 – Схема работы Sequence-to-Sequence модели с attention

Где h_i – i-ое скрытое состояние. $i \in (1,2,\ldots,m)$, а c_i обозначает i-й информационный вектор контекста, который вычисляется с помощью алгоритма attention.

Предложенная прогностическая модель предсказывает, какие системные вызовы будут выполнены впоследствии, что позволит наблюдать состояние системы и прогнозировать атаки.

2.1.2 LSTM

LSTM (Long Short-Term Memory) [9], [10], [11], [12] – это разновидность архитектуры рекуррентных нейронных сетей, которая способна запоминать информацию на длительные промежутки времени. LSTM состоит из специальных блоков памяти, которые позволяют сети запоминать и забывать информацию в зависимости от контекста.

Затухания градиентов – это явление, которое происходит при обучении нейронной сети, когда значения градиентов становятся очень маленькими по мере распространения через слои сети, что приводит к незначительным обновлениям весов, и обучение сети замедляется или останавливается.

Архитектура LSTM представлена на рис. 5.

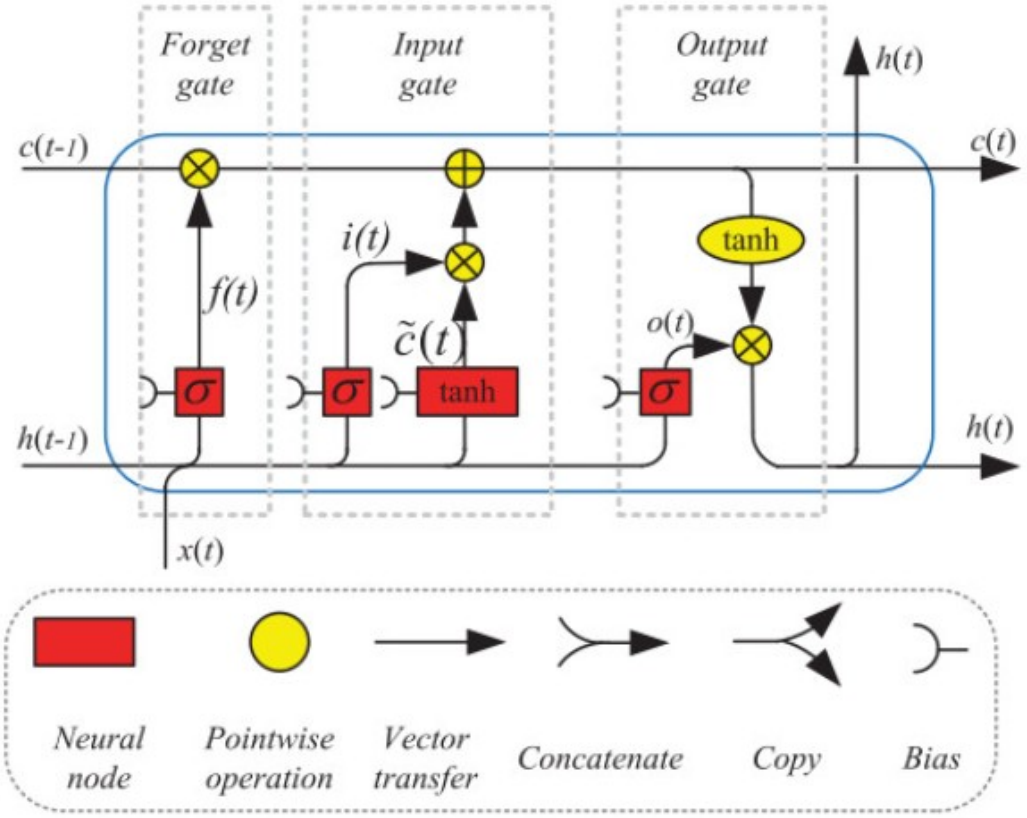

Рисунок 3 – Архитектура LSTM

Neural node – это обученный слой нейронной сети, pointwise operation – поточечная операция, vector transfer – векторный перенос (перемещение вектора с выхода одного узла на вход другого); concatenate – объединение; copy – копирование (информация копируется и отправляется в различные узлы сети), bias – добавление смещения.

В блоке памяти LSTM существует несколько основных «вентилей»:

1. Forget gate определяет, какую информацию можно забыть из предыдущей ячейки памяти. Помогает модели избегать проблемы затухания градиентов и сохранять только актуальную информацию.
2. Input gate решает, какую информацию следует добавить в ячейку памяти на основе нового входного сигнала. Позволяет модели обновлять свою память с учетом новых данных.
3. Output gate определяет, какую информацию из ячейки памяти следует передать на выход.

Сеть с двунаправленной LSTM (Bidirectional LSTM Network) [13], [14] – тип рекуррентной нейронной сети, который использует LSTM блоки и обрабатывает входные данные как в прямом, так и в обратном направлении. Этот подход позволяет учитывать как предыдущий, так и последующий контекст для более точного прогнозирования выходных данных.

Структура Bidirectional LSTM представлена на рис. 6.

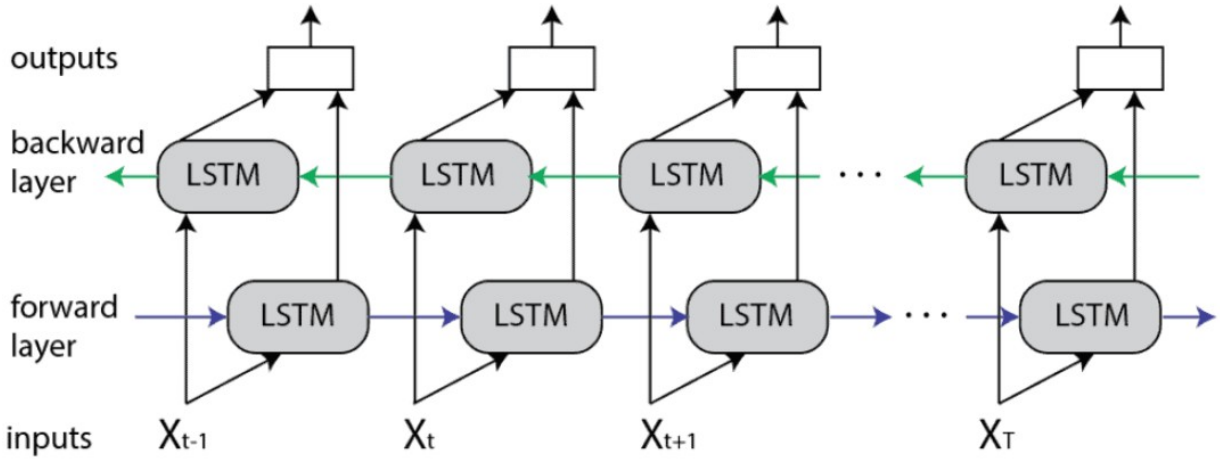

Рисунок 4 – Структура Bidirectional LSTM

Forward layer (прямой слой) работает в прямом направлении, начиная с первого элемента последовательности и последовательно обрабатывает следующие элементы до последнего.

Backward layer (обратный слой) работает в обратном направлении, начиная с последнего элемента последовательности и последовательно обрабатывает предыдущие элементы до первого.

Данный метод позволяет модели лучше прогнозировать следующий элемент на основе предыдущего и будущего контекста.

LSTM слои широко применяются для анализа и прогнозирования последовательных данных в различных областях науки и техники благодаря их способности эффективно работать с долгосрочными зависимостями в данных [15].

2.2 Критерии сравнения:

- **Точность прогнозирования:** Насколько точно модель предсказывает следующие системные вызовы (в процентах).
- **Учет двунаправленного контекста:** Способность модели учитывать контекст как предыдущих, так и последующих вызовов.
- **Вычислительная сложность:** Ресурсы, необходимые для обучения и использования модели.

2.3 Таблица сравнения:

2.4 Выводы по итогам сравнений: