Markov Model of Keyboard Tokens

0

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 Package components

Classes

- class Chain
- class Distribution
- class Token
- class Touch

  This class represents a touch event.
- class Window

  This class will store and provide functions for a single window within the model.

5.2 Package data_analysis

Classes

- class Model_compare
- class Model_compare_thread
- class Statistics

5.3 Package gui

Classes

- class Marcov_console_panel
- class Marcov_file_display_panel
- class Marcov_frame
- class Marcov_options_panel
- class StartGUI

5.4 Package junit

Classes

- class Unit_CompareChainsRank
- class Unit_CompleteProbability

5.5 Package rank

Classes

- class CompareChainsRank
- class CompleteProbability

5.6 Package runtime

Classes

- class ChainBuilder
- class CompareChains

  This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.
- class Operation_thread

5.7 Package test

Classes

- class Main
- class Print_model
- class UnitCompareChainsRank
- class UnitRankCompare

5.8 Package trie

Classes

- class Trie
- class TrieList

Chapter 6

Class Documentation

6.1 components.Chain Class Reference

Public Member Functions

- Chain (int window, int token, int threshold, int model_size)
- Chain (Chain c)

  copy constructor. New chain object should have the same state as the old with differant object references.
- void add_touch (Touch touch)
- void add_touch_list (List< Touch > t)
- void set_distribution (Distribution distribution, List< Distribution > key_distribution)
- double get_touch_probability (Window w, Touch t)

  returns the probability of a given touch (at the i'th index) based on the model. This will depend on the preceeding touches, in Window. A request for one probability will necessarily result in all of the probabilities being computed.
- Distribution get_distribution ()

  returns the distribution of the data as a whole
- List< Distribution > get_key_distribution ()

  returns a list of distributions for each key
- int get_window ()
- int get_token ()
- int get_model_size ()
- int get_threshold ()
- void reset ()

  resets the object.. this is the same as constructing a new chain, but faster
- void compute_uncomputed ()

  computes all uncomputed aspects of the chain
- double compare_to (Chain auth_chain)
- boolean is_touch_in_key_distribution (Touch touch)

  returns true if a touch is within 2 sigma for it's key distribution
- List< Window > get_windows ()

  handle requests for windows
- List< Token > get_tokens ()

  handle requests for tokens
- List< Touch > get_touches ()

  get a list of all touches in the chain
- String toString ()

  prints out all of the touches in order
- void output_to_csv (String file_name)

  NOT USEFUL IN ANDROID. This is used for debugging purposes. Outputs the model to a csv file in a readable format.

6.1.1 Detailed Description

TODO make the chain's compare_to method be able to update incrementally TODO make sure to use get_X↩XXXXX() instead of the instance variables TODO put windows into a Trie data structure for building model faster TODO anywhere where I need to compare windows, or Touches I need the option to do this with tokens there needs to be a way to set the distribution used for a chain. This is because the authentication chain is evaluated with the distribution of the base chain. compute the windows somewhere. This will be based on the threshold, window, token sizes. This may change distributions? if a touch is thrown out? this class represents the marcov chain. It contains a sequence of touches and a distribution. I avoid doing any processing on touch being added because eventually this will be called on key press in android. Setting it up this way is more flexible to in the sense that processing may be done at any time. caches the result of each computation so it does not have to be repeated.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 components.Chain.Chain (int *window,* int *token,* int *threshold,* int *model_size*)

6.1.2.2 components.Chain.Chain (Chain *c*)

copy constructor. New chain object should have the same state as the old with differant object references.

6.1.3 Member Function Documentation

6.1.3.1 void components.Chain.add_touch (Touch *touch*)

6.1.3.2 void components.Chain.add_touch_list (List< Touch > *t*)

6.1.3.3 double components.Chain.compare_to (Chain *auth_chain*)

returns the percent difference between this chain and auth_chain. the value returned will be between 0 and 1 0 indicates there is no difference 1 indicates there is a large difference

6.1.3.4 void components.Chain.compute_uncomputed ( )

computes all uncomputed aspects of the chain

6.1.3.5 Distribution components.Chain.get_distribution ( )

returns the distribution of the data as a whole

6.1.3.6 List<Distribution> components.Chain.get_key_distribution ( )

returns a list of distributions for each key

6.1.3.7 int components.Chain.get_model_size ( )

6.1.3.8 int components.Chain.get_threshold ( )

6.1.3.9 int components.Chain.get_token ( )

6.1.3.10 List< Token > components.Chain.get_tokens ( )

handle requests for tokens

6.1.3.11 double components.Chain.get_touch_probability (Window *w*, Touch *t*)

returns the probability of a given touch (at the i'th index) based on the model. This will depend on the preceeding touches, in Window. A request for one probability will necessarily result in all of the probabilities being computed.

6.1.3.12 List< Touch > components.Chain.get_touches ( )

get a list of all touches in the chain

6.1.3.13 int components.Chain.get_window ( )

6.1.3.14 List< Window > components.Chain.get_windows ( )

handle requests for windows

6.1.3.15 boolean components.Chain.is_touch_in_key_distribution (Touch *touch*)

returns true if a touch is within 2 sigma for it's key distribution

6.1.3.16 void components.Chain.output_to_csv (String *file_name*)

NOT USEFUL IN ANDROID. This is used for debugging purposes. Outputs the model to a csv file in a readable format.

6.1.3.17 void components.Chain.reset ( )

resets the object.. this is the same as constructing a new chain, but faster

6.1.3.18 void components.Chain.set_distribution (Distribution *distribution*, List< Distribution > *key_distribution*)

allows distribution to be set. If no distribution is set, the distribution for this chain of touches is computed. NOTE the distribution is not maintained when new touches are added.

6.1.3.19 String components.Chain.toString ( )

prints out all of the touches in order

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Chain.java

6.2 runtime.ChainBuilder Class Reference

Classes

- enum CompareMethod
- enum State

Public Member Functions

- ChainBuilder ()
- ChainBuilder (int window, int token, int threshold, int user_model_size, int auth_model_size)
- void handle_touch (Touch touch)
- void authenticate ()
- CompareChains get_authenticate_thread ()
- State get_authenticate_state ()

  handle requests for the current state of the authentication
- void build_chain_from_csv (File file)

Static Public Member Functions

- static List< Touch > parse_csv (File file)

  parse the csv file NOT USEFULL ON ANDROID

6.2.1 Detailed Description

TODO write a HashList class (most likely extends HashMap and implemetns list to store the hash of everything. Replace ArrayList with this class whereever arraylist is used. The other option is to use LinkedHashList. handles building of the model based on input events. This may not be necessary in android framework, but it will allow a consistant way of building the model across platforms to allow for easier migration to android device. whenever I add a touch, I take

6.2.2 Constructor & Destructor Documentation

6.2.2.1 runtime.ChainBuilder.ChainBuilder ( )

6.2.2.2 runtime.ChainBuilder.ChainBuilder (int *window,* int *token,* int *threshold,* int *user_model_size,* int *auth_model_size*)

allow model size, window, token values to be specified. This is mainly for testing purposes

6.2.3 Member Function Documentation

6.2.3.1 void runtime.ChainBuilder.authenticate ( )

allow forced authentication from outside of ChainBuilder. this involves starting the CompareChains. this method starts the authentication

6.2.3.2 void runtime.ChainBuilder.build_chain_from_csv (File *file*)

this code will NOT BE USEFULL ON ANDROID. It will build the model from a csv file in the current working directory. It will however utilize the handle_touch() method to add new touches to the chain. It is simply a matter of where the touches are coming from. TODO move this method to another place. it is only by convience that it exists here now.

6.2.3.3 State runtime.ChainBuilder.get_authenticate_state ( )

handle requests for the current state of the authentication

6.2.3.4 CompareChains runtime.ChainBuilder.get_authenticate_thread ( )

return the thread which is preforming the authentication. This method provides no guarentees about the state of the thread. It may even be null!

6.2.3.5 void runtime.ChainBuilder.handle_touch (Touch *touch*)

this method should be called in some way whenever there is a touch event in android. There should be minimal amounts of processing done here so the input to the device doesn't lag. I don't know by what method percicely this will need to be called in the android souce. It could be another class which simply handles touch events, or from the pre-existing android archetecture.

6.2.3.6 static List<Touch> runtime.ChainBuilder.parse_csv (File *file*) [static]

parse the csv file NOT USEFULL ON ANDROID

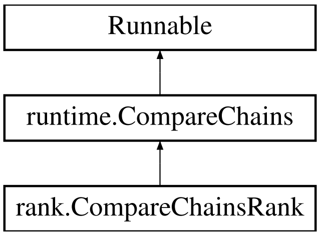
The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/ChainBuilder.java

6.3 runtime.CompareChains Class Reference

This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.

Inheritance diagram for runtime.CompareChains:



Public Member Functions

- CompareChains (Chain user_chain, Chain auth_chain)
- void run ()

  compare user_chain and auth_chain and choose what to do with the result
- double get_auth_probability ()

  returns the probability with which the
- boolean get_auth_result ()
- boolean get_auth_complete ()

Protected Attributes

- volatile boolean is_authentic
- volatile boolean complete
- Chain user_chain
- Chain auth_chain
- volatile double authentication_probability

6.3.1 Detailed Description

This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 runtime.CompareChains.CompareChains (Chain *user_chain,* Chain *auth_chain*)

will need to make copies of the chains passed in so they do not get updated by something else during the comparason

6.3.3 Member Function Documentation

6.3.3.1 boolean runtime.CompareChains.get_auth_complete ( )

6.3.3.2 double runtime.CompareChains.get_auth_probability ( )

returns the probability with which the

6.3.3.3 boolean runtime.CompareChains.get_auth_result ( )

returns the result of the authentication. This method does not provide any guarentees that the compairason has finsihed yet. If the compairason has not yet finished it will return false;

6.3.3.4 void runtime.CompareChains.run ( )

compare user_chain and auth_chain and choose what to do with the result

perform the comparison now that the values are cached in the Chain's

6.3.4 Member Data Documentation

6.3.4.1 Chain runtime.CompareChains.auth_chain [protected]

6.3.4.2 volatile double runtime.CompareChains.authentication_probability [protected]

6.3.4.3 volatile boolean runtime.CompareChains.complete [protected]

6.3.4.4 volatile boolean runtime.CompareChains.is_authentic [protected]

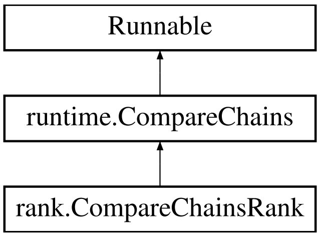
6.3.4.5 Chain runtime.CompareChains.user_chain [protected]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/CompareChains.java

6.4 rank.CompareChainsRank Class Reference

Inheritance diagram for rank.CompareChainsRank:



Public Member Functions

- CompareChainsRank (Chain user_chain, Chain auth_chain)
- void run ()

Additional Inherited Members

6.4.1 Constructor & Destructor Documentation

6.4.1.1 rank.CompareChainsRank.CompareChainsRank (Chain *user_chain,* Chain *auth_chain*)

6.4.2 Member Function Documentation

6.4.2.1 void rank.CompareChainsRank.run ( )

overrides the run method to implement the authentication with a page-rank style algorithm.

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/CompareChainsRank.java

6.5 runtime.ChainBuilder.CompareMethod Enum Reference

Public Attributes

- PROBABILITY_VECTOR_DIFFERANCE

6.5.1 Member Data Documentation

6.5.1.1 runtime.ChainBuilder.CompareMethod.PROBABILITY_VECTOR_DIFFERANCE

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/ChainBuilder.java

6.6 rank.CompleteProbability Class Reference

Public Member Functions

- CompleteProbability (Chain chain)
- Chain compute_probability ()

6.6.1 Detailed Description

This class computes probability in a different way from what is contained in the Chain class. This class looks at all of the touches to try to determine the probability that from any given touch, it transitions to another.

this is similar to having a window size of 1?

Author

element

6.6.2 Constructor & Destructor Documentation

6.6.2.1 rank.CompleteProbability.CompleteProbability (Chain *chain*)

6.6.3 Member Function Documentation

6.6.3.1 Chain rank.CompleteProbability.compute_probability ( )

make a replica of the chain with a window size of 1 and compute the probability.

Returns

replica chain

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/CompleteProbability.java

6.7 runtime.Operation_thread.Computation Enum Reference

Public Attributes

- DISTRIBUTION
- KEY_DISTRIBUTION
- WINDOW
- TOKEN
- PROBABILITY

6.7.1 Member Data Documentation

6.7.1.1 runtime.Operation_thread.Computation.DISTRIBUTION

6.7.1.2 runtime.Operation_thread.Computation.KEY_DISTRIBUTION

6.7.1.3 runtime.Operation_thread.Computation.PROBABILITY

6.7.1.4 runtime.Operation_thread.Computation.TOKEN

6.7.1.5 runtime.Operation_thread.Computation.WINDOW

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/Operation_thread.java

6.8 test.Main.TestFiles.Concentration Enum Reference

Public Member Functions

- Concentration (String description, int identifier, double value)
- String toString ()
- int get_identifier ()
- double get_value ()

Public Attributes

- HIGH =("High, [std deviation]", 0, 0)
- MEDIUM =("Medium, [std deviation]", 1, 0)
- LOW

6.8.1 Constructor & Destructor Documentation

6.8.1.1 test.Main.TestFiles.Concentration.Concentration (String *description,* int *identifier,* double *value*)

6.8.2 Member Function Documentation

6.8.2.1 int test.Main.TestFiles.Concentration.get_identifier ( )

6.8.2.2 double test.Main.TestFiles.Concentration.get_value ( )

6.8.2.3 String test.Main.TestFiles.Concentration.toString ( )

6.8.3 Member Data Documentation

6.8.3.1 test.Main.TestFiles.Concentration.HIGH =("High, [std deviation]", 0, 0)

6.8.3.2 test.Main.TestFiles.Concentration.LOW

Initial value:

```
=("Low, [std deviation]", 2,
                        0)
```

6.8.3.3 test.Main.TestFiles.Concentration.MEDIUM =("Medium, [std deviation]", 1, 0)

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java

6.9 test.Main.TestFiles.Distribution Enum Reference

Public Member Functions

- Distribution (String description, int identifier, double value)
- String toString ()
- int get_identifier ()
- double get_value ()

Public Attributes

- NORMAL =("Normal, centered about pressure median", 0, 0)
- ABNORMAL
- RANDOM =("Random, completly and utterly random", 2, 0)

6.9.1 Constructor & Destructor Documentation

6.9.1.1 test.Main.TestFiles.Distribution.Distribution (String *description,* int *identifier,* double *value*)

6.9.2 Member Function Documentation

6.9.2.1 int test.Main.TestFiles.Distribution.get_identifier ( )

6.9.2.2 double test.Main.TestFiles.Distribution.get_value ( )

6.9.2.3 String test.Main.TestFiles.Distribution.toString ( )

6.9.3 Member Data Documentation

6.9.3.1 test.Main.TestFiles.Distribution.ABNORMAL

Initial value:

```
=(
                        "Abnormal, centered about pressure median, but inverted", 1,
                        0)
```

6.9.3.2 test.Main.TestFiles.Distribution.NORMAL =("Normal, centered about pressure median", 0, 0)

6.9.3.3 test.Main.TestFiles.Distribution.RANDOM =("Random, completly and utterly random", 2, 0)

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java

6.10 components.Distribution Class Reference

Public Member Functions

- Distribution (List< Touch > touches)
- Distribution (List< Touch > touches, int keycode)

  this constructor allows a keycode to be associated with the distribution
- Distribution (Distribution d)

  copy constructor. This exists because computations are done in the constructor. Copying in this way avoids recomputation.
- void update (List< Touch > touches)

  updates the distribution using a list of touches. This update has nothing to do with the old values in the distribution. It is synonomous to creating a new Distribution object with this list of touches.
- double get_min ()
- double get_max ()
- double get_average ()
- double get_standard_deviation ()
- int get_keycode ()

  returns the keycode associated with this distribution. If the distribution does not have an associated keycode, this method will return -1.
- boolean equals (Object o)

  determine if this distribution is exactly equal to another distribution

6.10.1 Detailed Description

TODO make the computations happen at request time, and cache the result so it does not need to be recomputed. Or leave it as is.... as distribution objects are only created as needed in the rest of the code. this class knows how to calculate the distribution of a list of touches

6.10.2 Constructor & Destructor Documentation

6.10.2.1 components.Distribution.Distribution (List< **Touch** > *touches*)

6.10.2.2 components.Distribution.Distribution (List< **Touch** > *touches,* int *keycode*)

this constructor allows a keycode to be associated with the distribution

6.10.2.3 components.Distribution.Distribution (**Distribution** *d*)

copy constructor. This exists because computations are done in the constructor. Copying in this way avoids recomputation.

6.10.3 Member Function Documentation

6.10.3.1 boolean components.Distribution.equals (Object *o*)

determine if this distribution is exactly equal to another distribution

6.10.3.2 double components.Distribution.get_average ( )

6.10.3.3 int components.Distribution.get_keycode ( )

returns the keycode associated with this distribution. If the distribution does not have an associated keycode, this method will return -1.

6.10.3.4 double components.Distribution.get_max ( )

6.10.3.5 double components.Distribution.get_min ( )

6.10.3.6 double components.Distribution.get_standard_deviation ( )

6.10.3.7 void components.Distribution.update (List< Touch > *touches*)

updates the distribution using a list of touches. This update has nothing to do with the old values in the distribution. It is synonomous to creating a new Distribution object with this list of touches.

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Distribution.java

6.11 test.Main Class Reference

Static Public Member Functions

- static void main (String args[ ])

6.11.1 Detailed Description

TODO generate a csv file for testing This class is used to test that the model is being built correctly. Also tested is the model compairason. and various classes used in model creating. The idea is to print out the tests which fail. This class should have to do no actual work if the program is designed well.

6.11.2 Member Function Documentation

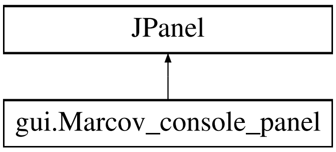
6.11.2.1 static void test.Main.main (String *args[ ]*) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java

6.12 gui.Marcov_console_panel Class Reference

Inheritance diagram for gui.Marcov_console_panel:



Public Member Functions

- Marcov_console_panel ()

6.12.1 Constructor & Destructor Documentation

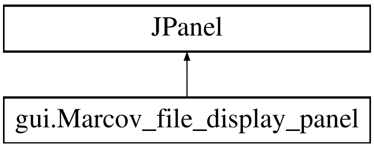
6.12.1.1 gui.Marcov_console_panel.Marcov_console_panel ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_console_panel.java

6.13 gui.Marcov_file_display_panel Class Reference

Inheritance diagram for gui.Marcov_file_display_panel:



Public Member Functions

- Marcov_file_display_panel ()

6.13.1 Constructor & Destructor Documentation

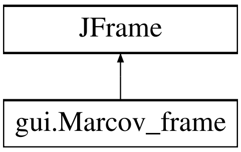
6.13.1.1 gui.Marcov_file_display_panel.Marcov_file_display_panel ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_file_display_panel.java

6.14 gui.Marcov_frame Class Reference

Inheritance diagram for gui.Marcov_frame:



Public Member Functions

- Marcov_frame ()
- void close ()

6.14.1 Constructor & Destructor Documentation

6.14.1.1 gui.Marcov_frame.Marcov_frame ( )

6.14.2 Member Function Documentation

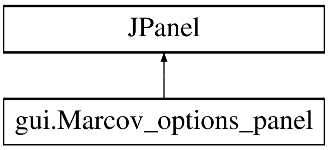
6.14.2.1 void gui.Marcov_frame.close ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_frame.java

6.15 gui.Marcov_options_panel Class Reference

Inheritance diagram for gui.Marcov_options_panel:



Public Member Functions

- Marcov_options_panel ()

6.15.1 Constructor & Destructor Documentation

6.15.1.1 gui.Marcov_options_panel.Marcov_options_panel ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_options_panel.java

6.16 data_analysis.Model_compare Class Reference

Static Public Member Functions

- static void main (String[ ] args)

6.16.1 Detailed Description

The purpose of this class is to test out the model compare process on data that has been collected The data to used will be contained in the data_sets folder input: data_sets folder output: model_compare_output.txt

6.16.2 Member Function Documentation

6.16.2.1 static void data_analysis.Model_compare.main (String[ ] *args*) [static]

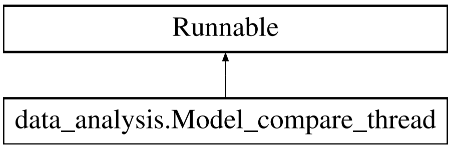
create a number of tests with different parameters

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Model_compare.java

6.17 data_analysis.Model_compare_thread Class Reference

Inheritance diagram for data_analysis.Model_compare_thread:



Public Member Functions

- Model_compare_thread (String base_data_path, String auth_data_path, int base_model_size, int auth_model_size, int window_size, int token_size, int threshold)

  constructor, allowing user to set different probperties of the model compairason for testing
- void run ()
- String get_base_data_path ()
- String get_auth_data_path ()
- int get_window_size ()
- int get_token_size ()
- int get_threshold ()
- int get_base_model_size ()
- int get_auth_model_size ()
- List< Double > get_auth_probability_list ()

Public Attributes

- double max_authentication_probability
- double min_authentication_probability
- double average_authentication_probability

6.17.1 Detailed Description

this thread allows the preforming of a test compairason. when the compairason is finished, an instance variable will be set indicating different results.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 data_analysis.Model_compare_thread.Model_compare_thread (String *base_data_path,* String *auth_data_path,* int *base_model_size,* int *auth_model_size,* int *window_size,* int *token_size,* int *threshold*)

constructor, allowing user to set different probperties of the model compairason for testing

6.17.3 Member Function Documentation

6.17.3.1 String data_analysis.Model_compare_thread.get_auth_data_path ( )

6.17.3.2 int data_analysis.Model_compare_thread.get_auth_model_size ( )

6.17.3.3 List<Double> data_analysis.Model_compare_thread.get_auth_probability_list ( )

6.17.3.4 String data_analysis.Model_compare_thread.get_base_data_path ( )

6.17.3.5 int data_analysis.Model_compare_thread.get_base_model_size ( )

6.17.3.6 int data_analysis.Model_compare_thread.get_threshold ( )

6.17.3.7 int data_analysis.Model_compare_thread.get_token_size ( )

6.17.3.8 int data_analysis.Model_compare_thread.get_window_size ( )

6.17.3.9 void data_analysis.Model_compare_thread.run ( )

6.17.4 Member Data Documentation

6.17.4.1 double data_analysis.Model_compare_thread.average_authentication_probability

6.17.4.2 double data_analysis.Model_compare_thread.max_authentication_probability

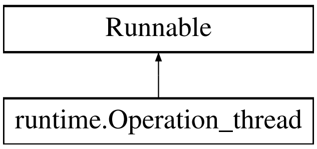
6.17.4.3 double data_analysis.Model_compare_thread.min_authentication_probability

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Model_compare_↩ thread.java

6.18 runtime.Operation_thread Class Reference

Inheritance diagram for runtime.Operation_thread:



Classes

- enum Computation

Public Member Functions

- Operation_thread (Chain chain, Computation computation)
- void run ()

6.18.1 Constructor & Destructor Documentation

6.18.1.1 runtime.Operation_thread.Operation_thread (Chain *chain,* Computation *computation*)

6.18.2 Member Function Documentation

6.18.2.1 void runtime.Operation_thread.run ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/Operation_thread.java

6.19 test.Main.TestFiles.PressureAmount Enum Reference

Public Member Functions

- PressureAmount (String description, int identifier, double value)
- String toString ()
- int get_identifier ()
- double get_value ()

Public Attributes

- HIGH =("High pressure, 0.75", 0, .75)
- MEDIUM =("Medium Pressure, 0.5", 1, .5)
- LOW

6.19.1 Constructor & Destructor Documentation

6.19.1.1 test.Main.TestFiles.PressureAmount.PressureAmount (String *description,* int *identifier,* double *value*)

6.19.2 Member Function Documentation

6.19.2.1 int test.Main.TestFiles.PressureAmount.get_identifier ( )

6.19.2.2 double test.Main.TestFiles.PressureAmount.get_value ( )

6.19.2.3 String test.Main.TestFiles.PressureAmount.toString ( )

6.19.3 Member Data Documentation

6.19.3.1 test.Main.TestFiles.PressureAmount.HIGH =("High pressure, 0.75", 0, .75)

6.19.3.2 test.Main.TestFiles.PressureAmount.LOW

Initial value:

```
=("Low Pressure, 0.25", 2,
                          .25)
```

6.19.3.3 test.Main.TestFiles.PressureAmount.MEDIUM =("Medium Pressure, 0.5", 1, .5)

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java

6.20 test.Print_model Class Reference

Static Public Member Functions

- static void main (String[ ] args)

6.20.1 Member Function Documentation

6.20.1.1 static void test.Print_model.main (String[ ] *args*) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Print_model.java

6.21 gui.StartGUI Class Reference

Static Public Member Functions

- static void main (String[ ] args)
- static void exit ()

  causes the frame to close

6.21.1 Member Function Documentation

6.21.1.1 static void gui.StartGUI.exit ( ) [static]

causes the frame to close

6.21.1.2 static void gui.StartGUI.main (String[ ] *args*) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/StartGUI.java

6.22 runtime.ChainBuilder.State Enum Reference

Public Attributes

- IN_PROGRESS
- SUCCESS

6.22.1 Member Data Documentation

6.22.1.1 runtime.ChainBuilder.State.IN_PROGRESS

6.22.1.2 runtime.ChainBuilder.State.SUCCESS

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/ChainBuilder.java

6.23 data_analysis.Statistics Class Reference

Static Public Member Functions

- static void main (String args[ ])
- static double false_positive_percentage (double authentication_percentage, List< Double > should_↩authenticate_percentages, List< Double > should_not_authenticate_percentages)
- static double false_negative_percentage (double authentication_percentage, List< Double > should_↩authenticate_percentages, List< Double > should_not_authenticate_percentages)
- static double best_authentication_percentage (List< Double > should_authenticate_percentages, List< Double > should_not_authenticate_percentages)
- static double minimize_false_positive_authentication_percentage (List< Double > should_authenticate_↩percentages, List< Double > should_not_authenticate_percentages)
- static double equal_false_positive_negative_authentication_percentage (List< Double > should_↩authenticate_percentages, List< Double > should_not_authenticate_percentages)
- static double authentication_accuracy (double authentication_percentage, List< Double > should_↩authenticate_percentages, List< Double > should_not_authenticate_percentages)

6.23.1 Member Function Documentation

6.23.1.1 static double data_analysis.Statistics.authentication_accuracy (double *authentication_percentage,* List< Double > *should_authenticate_percentages,* List< Double > *should_not_authenticate_percentages*) [static]

6.23.1.2 static double data_analysis.Statistics.best_authentication_percentage (List< Double > *should_authenticate_percentages,* List< Double > *should_not_authenticate_percentages*) [static]

6.23.1.3 static double data_analysis.Statistics.equal_false_positive_negative_authentication_percentage (List< Double > *should_authenticate_percentages,* List< Double > *should_not_authenticate_percentages*) [static]

6.23.1.4 static double data_analysis.Statistics.false_negative_percentage (double *authentication_percentage,* List< Double > *should_authenticate_percentages,* List< Double > *should_not_authenticate_percentages*) [static]

6.23.1.5 static double data_analysis.Statistics.false_positive_percentage (double *authentication_percentage,* List< Double > *should_authenticate_percentages,* List< Double > *should_not_authenticate_percentages*) [static]

6.23.1.6 static void data_analysis.Statistics.main (String *args[ ]*) [static]

6.23.1.7 static double data_analysis.Statistics.minimize_false_positive_authentication_percentage (List< Double > *should_authenticate_percentages,* List< Double > *should_not_authenticate_percentages*) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Statistics.java

6.24 components.Token Class Reference

Classes

- enum Type

  specify the type of token we want to build

Public Member Functions

- Token (Distribution distribution, int total_tokens, int token_index, double standard_deviations, Type type)

  allow tokens to be created making each touch mu for its keycode, or in a linear fashion over the distribution for the keycode
- Token (Distribution distribution, int total_tokens, int token_index, Type type)

  allow for creation of tokens over a distribution
- Token (double range_min, double range_max, int total_tokens, int token_index, Type type)
- boolean contains (Touch touch)
- boolean is_high_wildcard (Touch touch)

  determine if a touch is a high_wildcard
- boolean is_low_wildcard (Touch touch)

  determine if a touch is a low wildcard
- void increment_high_wildcards ()

  adds the the number of high wildcards
- void increment_low_wildcards ()

  adds the the number of high wildcards
- int get_total_wildcards ()

  returns the total number of wildcards
- int get_acceptable_wildcards (int total_items)
- double get_min ()

  returns true if there are more than an acceptable number of wildcards
- double get_max ()

  return the maximum
- boolean equals (Object o_t)

  compares This token to another to determine if they are the same

6.24.1 Detailed Description

This class represents a token within the model. Essentially this is a range of values. A touch is defined to be within a token if the pressure value of the touch falls within this range. This class is designed to abstract away the clustering algorithm. This makes the rest of the code far simpler to think about Something to look at in the future may be a clustering algorithm that is not equally distributed

6.24.2 Constructor & Destructor Documentation

6.24.2.1 components.Token.Token (Distribution *distribution,* int *total_tokens,* int *token_index,* double *standard_deviations,* Type *type*)

allow tokens to be created making each touch mu for its keycode, or in a linear fashion over the distribution for the keycode

allow for creation of tokens with-in some number of standard deviations of a distribution

6.24.2.2 components.Token.Token (Distribution *distribution,* int *total_tokens,* int *token_index,* Type *type*)

allow for creation of tokens over a distribution

6.24.2.3 components.Token.Token (double *range_min,* double *range_max,* int *total_tokens,* int *token_index,* Type *type*)

Implemented in the constructor is the clustering algorithm. This determines how to split up the range into a number of tokens.

range_min, minimum of the token range range_max, maximum of the token range total_tokens total number of tokens to split range into token_index, integer between 0 and total_tokens-1 indicating into which range this token falls

6.24.3 Member Function Documentation

6.24.3.1 boolean components.Token.contains (Touch *touch*)

determines if a touch is within this token based on its pressure value this will return true if a touches pressure equals max or min, so if max of one token is min of another token, both will return true

6.24.3.2 boolean components.Token.equals (Object *o_t*)

compares This token to another to determine if they are the same

6.24.3.3 int components.Token.get_acceptable_wildcards (int *total_items*)

returns the acceptable number of wildcards

Parameters

total_items	is the total number of items in the distribution

6.24.3.4 double components.Token.get_max ( )

return the maximum

6.24.3.5 double components.Token.get_min ( )

returns true if there are more than an acceptable number of wildcards

return minimum

6.24.3.6 int components.Token.get_total_wildcards ( )

returns the total number of wildcards

6.24.3.7 void components.Token.increment_high_wildcards ( )

adds the the number of high wildcards

6.24.3.8 void components.Token.increment_low_wildcards ( )

adds the the number of high wildcards

6.24.3.9 boolean components.Token.is_high_wildcard (Touch *touch*)

determine if a touch is a high_wildcard

6.24.3.10 boolean components.Token.is_low_wildcard (Touch *touch*)

determine if a touch is a low wildcard

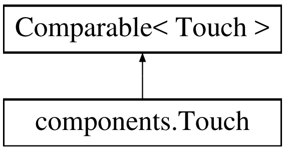
The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Token.java

6.25 components.Touch Class Reference

This class represents a touch event.

Inheritance diagram for components.Touch:



Public Member Functions

- Touch (int keycode, double pressure, long timestamp)
- Touch (Touch t)

    copy constructor
- void set_probability (Window preceeding_window, double p)

    sets the probability that this touch succeeds a given sequence. Reccord the sequence and the probability
- double get_probability (Window preceeding_window)

    returns the probability of the touch occurring after a given window w. If the window does not exist return (TODO) currently returning 0
- double get_pressure ()
- int get_key ()
- long get_timestamp ()
- boolean compare_with_token (List< Token > tokens, Touch other_touch)
- int hashCode ()

    implement hash function for the touch class
- int compareTo (Touch other_touch)

    compare touches to one another. return negative if this touch is less than other_touch
- String toString ()

6.25.1 Detailed Description

This class represents a touch event.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 components.Touch.Touch (int *keycode,* double *pressure,* long *timestamp*)

6.25.2.2 components.Touch.Touch (Touch *t*)

copy constructor

6.25.3 Member Function Documentation

6.25.3.1 boolean components.Touch.compare_with_token (List< Token > *tokens,* Touch *other_touch*)

compares the touches with the given token list. this function will return true if the touches are contained within the smae token

6.25.3.2 int components.Touch.compareTo (Touch *other_touch*)

compare touches to one another. return negative if this touch is less than other_touch

6.25.3.3 int components.Touch.get_key ( )

6.25.3.4 double components.Touch.get_pressure ( )

6.25.3.5 double components.Touch.get_probability (Window *preceeding_window*)

returns the probability of the touch occurring after a given window w. If the window does not exist return (TODO) currently returning 0

6.25.3.6 long components.Touch.get_timestamp ( )

6.25.3.7 int components.Touch.hashCode ( )

implement hash function for the touch class

6.25.3.8 void components.Touch.set_probability (Window *preceeding_window*, double *p*)

sets the probability that this touch succeeds a given sequence. Reccord the sequence and the probability

6.25.3.9 String components.Touch.toString ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Touch.java

6.26 trie.Trie Class Reference

Classes

- class **TrieNode**

Public Member Functions

- Trie ()

  sets up the tree so that everything will be added to trienode root?
- Trie (Trie t)

  creates a copy trie
- void clear ()

  removes all elements from the trie
- void insertString (String s, int index)
- int occurrence_count (String s)

  retrieves the number of occurrences of a given string in the tree
- List< Integer > get_index_list (String s)

  returns a list of indexes containing the given window
- void printSorted (TrieNode node, String s)

  prints the elements in a sorted order

6.26.1 Detailed Description

Credit for portions of this implementation to: `https://sites.google.com/site/indy256/algo/trie`

6.26.2 Constructor & Destructor Documentation

6.26.2.1 trie.Trie.Trie ( )

sets up the tree so that everything will be added to trienode root?

6.26.2.2 trie.Trie.Trie (Trie *t*)

creates a copy trie

6.26.3 Member Function Documentation

6.26.3.1 void trie.Trie.clear ( )

removes all elements from the trie

6.26.3.2 List<Integer> trie.Trie.get_index_list (String *s*)

returns a list of indexes containing the given window

6.26.3.3 void trie.Trie.insertString (String *s,* int *index*)

6.26.3.4 int trie.Trie.occurrence_count (String *s*)

retrieves the number of occurrences of a given string in the tree

6.26.3.5 void trie.Trie.printSorted (TrieNode *node,* String *s*)

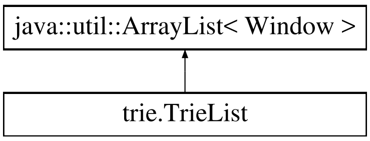
prints the elements in a sorted order

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/Trie.java

6.27 trie.TrieList Class Reference

Inheritance diagram for trie.TrieList:



Public Member Functions

- TrieList ()
- TrieList (TrieList t)
- boolean add (Window arg0)
- void add (int arg0, Window arg1)
- boolean addAll (Collection< ?extends Window > arg0)
- boolean addAll (int arg0, Collection< ?extends Window > arg1)
- void clear ()
- boolean remove (Object arg0)
- Window remove (int arg0)
- boolean removeAll (Collection< ?> arg0)
- boolean retainAll (Collection< ?> arg0)
- Window set (int arg0, Window arg1)
- int successor_count (List< Touch > successor_list, Window window, Touch touch)

    counts the number of times a given touch comes after a given window. in the given window, succesors list
- int occurrence_count (Window w)
- void set_tokens (List< Token > tokens)

    sets the tokens that will be used when encoding the window

6.27.1 Detailed Description

TODO Eventually this will be implemented as a prefix tree. This will greatly speed up many of the operations causing the calculation of the probabilities to be slow. right now it is fine to have the backing be an arraylist overrided methods are any that remove, modify, or add to the arraylist. these methods will also change the prefix tree

6.27.2 Constructor & Destructor Documentation

6.27.2.1 trie.TrieList.TrieList ( )

6.27.2.2 trie.TrieList.TrieList (TrieList *t*)

6.27.3 Member Function Documentation

6.27.3.1 boolean trie.TrieList.add (Window *arg0*)

6.27.3.2 void trie.TrieList.add (int *arg0*, Window *arg1*)

6.27.3.3 boolean trie.TrieList.addAll (Collection< ?extends **Window** > *arg0*)

6.27.3.4 boolean trie.TrieList.addAll (int *arg0*, Collection< ?extends **Window** > *arg1*)

6.27.3.5 void trie.TrieList.clear ( )

6.27.3.6 int trie.TrieList.occurrence_count (**Window** *w*)

return the number of occurrences of w in window_list TODO I think this method needs to be faster. Storing windows in a prefix tree would allow for this

6.27.3.7 boolean trie.TrieList.remove (Object *arg0*)

6.27.3.8 **Window** trie.TrieList.remove (int *arg0*)

6.27.3.9 boolean trie.TrieList.removeAll (Collection< ?> *arg0*)

6.27.3.10 boolean trie.TrieList.retainAll (Collection< ?> *arg0*)

6.27.3.11 **Window** trie.TrieList.set (int *arg0*, **Window** *arg1*)

6.27.3.12 void trie.TrieList.set_tokens (List< **Token** > *tokens*)

sets the tokens that will be used when encoding the window

6.27.3.13 int trie.TrieList.successor_count (List< **Touch** > *successor_list*, **Window** *window*, **Touch** *touch*)

counts the number of times a given touch comes after a given window. in the given window, succesors list

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/TrieList.java

6.28 components.Token.Type Enum Reference

specify the type of token we want to build

Public Attributes

- linear
- keycode_mu
- combined

6.28.1 Detailed Description

specify the type of token we want to build

6.28.2 Member Data Documentation

6.28.2.1 components.Token.Type.combined

6.28.2.2 components.Token.Type.keycode_mu

6.28.2.3 components.Token.Type.linear

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Token.java

6.29 junit.Unit_CompareChainsRank Class Reference

Public Member Functions

- void init ()
- void test_authentication_probability ()

6.29.1 Detailed Description

goal is to test compare chains rank functionality

6.29.2 Member Function Documentation

6.29.2.1 void junit.Unit_CompareChainsRank.init ( )

6.29.2.2 void junit.Unit_CompareChainsRank.test_authentication_probability ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_CompareChainsRank.java

6.30 junit.Unit_CompleteProbability Class Reference

Public Member Functions

- void init ()
- void test_replica_distribution ()

6.30.1 Member Function Documentation

6.30.1.1 void junit.Unit_CompleteProbability.init ( )

6.30.1.2 void junit.Unit_CompleteProbability.test_replica_distribution ( )

test different properties of replica chain to see if this works as expected. Replica chain should contain the probabilities for when window is equal to 1.

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_CompleteProbability.java

6.31 test.UnitCompareChainsRank Class Reference

Public Member Functions

- void init ()
- void test_chain_to_graph ()
- void test_touch_index ()
- void test_touch_window ()
- void test ()

6.31.1 Member Function Documentation

6.31.1.1 void test.UnitCompareChainsRank.init ( )

6.31.1.2 void test.UnitCompareChainsRank.test ( )

example test

6.31.1.3 void test.UnitCompareChainsRank.test_chain_to_graph ( )

tests chain_to_graph method to make sure the chain is converted to a StateGraph correctly.

6.31.1.4 void test.UnitCompareChainsRank.test_touch_index ( )

make sure touch index returns the correct index in the list

6.31.1.5 void test.UnitCompareChainsRank.test_touch_window ( )

make sure the touch_window() returns the correct window in chain.

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitCompareChainsRank.java

6.32 test.UnitRankCompare Class Reference

Public Member Functions

- void init ()
- void test_compare_correct ()
- void test_auth_probability ()
- void test ()

6.32.1 Detailed Description

Test the compairason with ranks

Author

element

6.32.2 Member Function Documentation

6.32.2.1 void test.UnitRankCompare.init ( )

6.32.2.2 void test.UnitRankCompare.test ( )

example test

6.32.2.3 void test.UnitRankCompare.test_auth_probability ( )

check the test that the compare vectors are correct

6.32.2.4 void test.UnitRankCompare.test_compare_correct ( )

checks that the probabilities are correct

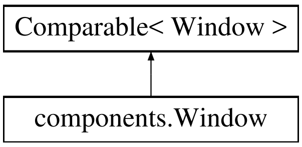
The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitRankCompare.java

6.33 components.Window Class Reference

This class will store and provide functions for a single window within the model.

Inheritance diagram for components.Window:



Public Member Functions

- Window (List< Touch > touches)
- Window (Window w)

  copy constructor
- boolean compare_with_token (List< Token > tokens, Window other_window)
- int size ()

  returns the number of touches in the window
- List< Touch > get_touch_list ()

  returns the window in the form of a touch list
- int hashCode ()

  implement a hash function which returns the hash of the current window
- int compareTo (Window other_window)

  compare this window to another window. Return negative if this window is less than the other window. Comparason is based on touches' pressure. Returns 0 if they are equal.
- String toString ()

6.33.1 Detailed Description

This class will store and provide functions for a single window within the model.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 components.Window.Window (List< **Touch** > *touches*)

6.33.2.2 components.Window.Window (**Window** *w*)

copy constructor

6.33.3 Member Function Documentation

6.33.3.1 boolean components.Window.compare_with_token (List< **Token** > *tokens,* **Window** *other_window*)

used for compairason of windows with a given token set. return true if this window is equal to auth window.

6.33.3.2 int components.Window.compareTo (**Window** *other_window*)

compare this window to another window. Return negative if this window is less than the other window. Comparason is based on touches' pressure. Returns 0 if they are equal.

6.33.3.3 List<**Touch**> components.Window.get_touch_list ( )

returns the window in the form of a touch list

6.33.3.4 int components.Window.hashCode ( )

implement a hash function which returns the hash of the current window

6.33.3.5 int components.Window.size ( )

returns the number of touches in the window

6.33.3.6 String components.Window.toString ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Window.java

Chapter 7

File Documentation

7.1 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/↩ Chain.java File Reference

Classes

- class components.Chain

Packages

- package components

7.2 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/↩ Distribution.java File Reference

Classes

- class components.Distribution

Packages

- package components

7.3 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/↩ Token.java File Reference

Classes

- class components.Token
- enum components.Token.Type

  specify the type of token we want to build

Packages

- package components

7.4 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/↩ Touch.java File Reference

Classes

- class components.Touch

  This class represents a touch event.

Packages

- package components

7.5 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/↩ Window.java File Reference

Classes

- class components.Window

  This class will store and provide functions for a single window within the model.

Packages

- package components

7.6 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/↩ Model_compare.java File Reference

Classes

- class data_analysis.Model_compare

Packages

- package data_analysis

7.7 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/↩ Model_compare_thread.java File Reference

Classes

- class data_analysis.Model_compare_thread

Packages

- package data_analysis

7.8 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/↩ Statistics.java File Reference

Classes

- class data_analysis.Statistics

Packages

- package data_analysis

7.9 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_↩ console_panel.java File Reference

Classes

- class gui.Marcov_console_panel

Packages

- package gui

7.10 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_↩ file_display_panel.java File Reference

Classes

- class gui.Marcov_file_display_panel

Packages

- package gui

7.11 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_↩ frame.java File Reference

Classes

- class gui.Marcov_frame

Packages

- package gui

7.12 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_↩ options_panel.java File Reference

Classes

- class gui.Marcov_options_panel

Packages

- package gui

7.13 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/StartG↩ UI.java File Reference

Classes

- class gui.StartGUI

Packages

- package gui

7.14 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_↩ CompareChainsRank.java File Reference

Classes

- class junit.Unit_CompareChainsRank

Packages

- package junit

7.15 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_↩ CompleteProbability.java File Reference

Classes

- class junit.Unit_CompleteProbability

Packages

- package junit

7.16 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/Compare↩ ChainsRank.java File Reference

Classes

- class rank.CompareChainsRank

Packages

- package rank

7.17 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/Complete↩ Probability.java File Reference

Classes

- class rank.CompleteProbability

Packages

- package rank

7.18 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/Chain↩ Builder.java File Reference

Classes

- class runtime.ChainBuilder
- enum runtime.ChainBuilder.State
- enum runtime.ChainBuilder.CompareMethod

Packages

- package runtime

7.19 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/Compare↩ Chains.java File Reference

Classes

- class runtime.CompareChains

  This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.

Packages

- package runtime

7.20 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/Operation↩ _thread.java File Reference

Classes

- class runtime.Operation_thread
- enum runtime.Operation_thread.Computation

Packages

- package runtime

7.21 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java File Reference

Classes

- class test.Main
- enum test.Main.TestFiles.PressureAmount
- enum test.Main.TestFiles.Distribution
- enum test.Main.TestFiles.Concentration

Packages

- package test

7.22 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Print_↩ model.java File Reference

Classes

- class test.Print_model

Packages

- package test

7.23 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitCompare↩ ChainsRank.java File Reference

Classes

- class test.UnitCompareChainsRank

Packages

- package test

7.24 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitRank↩ Compare.java File Reference

Classes

- class test.UnitRankCompare

Packages

- package test

7.25 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Utilities.java File Reference

7.26 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/Trie.java File Reference

Classes

- class trie.Trie
- class **trie.Trie.TrieNode**

Packages

- package trie

7.27 /home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/TrieList.java File Reference

Classes

- class trie.TrieList

Packages

- package trie

Index