CLAIMS:

1. A computer server for detecting policy violating applications comprising:

a processor; and

a non-transitory media readable by the processor, the media storing instructions that when executed by the processor, cause the processor to:

obtain bad applications from an application store;

generate a feature table based on identities of the bad applications, a blacklist, a permission list, an Application Programming Interface (API) list and a metadata list;

obtain parameters for the bad applications using the feature table;

obtain parameters for good applications using the feature table;

generate a training dataset based on the bad applications and the parameters for the bad applications, and the good applications and the parameters for the good applications; and

train a supervised machine learning algorithm using the training dataset, wherein the trained supervised machine learning algorithm is used to detect policy violating applications.

2. The computer server according to claim 1 wherein the instructions to obtain bad applications from the application store comprises:

instructions for directing the processor to:

retrieve applications from the application store;

determine applications that violate policies of the application store, and classify applications that violate policies of the application store as bad applications.

3. The computer server according to claim 2 wherein the instructions to retrieve the applications from the application store comprises:

instructions for directing the processor to:

obtain a listing of applications from an external source; and

retrieve applications contained in the listing of applications from the application store.

4. The computer server according to any one of claims 2 or 3 wherein the instructions to determine the applications that violate the policies of the application store comprises:

instructions for directing the processor to:

determine, after a period of time, if the retrieved applications are contained in the application store, wherein retrieved applications that are not contained in the application store after the period of time are classified as bad applications.

5. The computer server according to any one of claims 1-4 wherein the instructions to generate the feature table comprises:

instructions for directing the processor to:

compile the blacklist based on a set of common terms, Intellectual Property (IP) violating terms, third-party libraries having aggressive advertisement behaviour and identities of undesirable computer servers;

compile the permission list based on permissions that access secure device data and device access control data;

compile the API call list based on a set of predetermined API calls and obfuscated API calls;

compile the metadata list based on a set of predetermined metadata,

whereby the blacklist, the permission list, the API call list and the metadata list are each assigned a column in the feature table and each bad application is assigned a row in the feature table.

6. The computer server according to any one of claims 1-5 wherein the instructions to obtain parameters for bad applications using the feature table comprises:

instructions for directing the processor to:

for each bad application contained in the feature table,

set a blacklist parameter using identified extracted features;

set a permission parameter using determined extracted features;

set an API call parameter using established extracted features; and

set a metadata parameter using discerned extracted features.

7. The computer server according to claim 6 wherein the instructions to set the blacklist parameter using the identified extracted features comprises:

instructions for directing the processor to:

extract features from the bad application;

identify terms and libraries from the extracted features that match terms and libraries contained within the blacklist;

identify package files from the extracted features that interact with servers contained in the blacklist; and

set the blacklist parameter using the identified extracted features.

8. The computer server according to claim 6 or 7 wherein the instructions to set the permission parameter using the determined extracted features comprises:

instructions for directing the processor to:

extract features from the bad application;

determine permissions from the extracted features that match permissions contained within the permission list; and

set the permission parameter using the determined extracted features.

9. The computer server according to any one of claim 6-8 wherein the instructions to set the API call parameter using the established extracted features comprises:

instructions for directing the processor to:

extract features from the bad application;

establish API calls from the extracted features that match API calls contained within the API call list; and

set the API call parameter using the established API calls.

10. The computer server according to any one of claims 6-9 wherein the instructions to set the metadata parameter using the discerned extracted features comprises:

instructions for directing the processor to:

extract features from the bad application;

discern metadata from the extracted features that comply with metadata queries contained within the metadata list; and

set the metadata parameter using the discerned metadata.

11. The computer server according to any one of claims 6 to 10 wherein the extracted features from the bad applications comprise brand names, IP terms, network activities, third-party libraries, permissions, API calls or metadata.

12. The computer server according to any one of claims 1-11 wherein the supervised machine learning algorithm comprises algorithms based on regression models, support vector machine models, K-nearest neighbour models, base classifier models, simple decision tree models or decision tree models.

13. A method for detecting policy violating applications using a computer server comprising:

obtaining bad applications from an application store;

generating a feature table based on identities of the bad applications, a blacklist, a permission list, an Application Programming Interface (API) list and a metadata list;

obtaining parameters for the bad applications using the feature table;

obtaining parameters for good applications using the feature table;

generating a training dataset based on the bad applications and the parameters for the bad applications, and the good applications and the parameters for the good applications; and

training a supervised machine learning algorithm using the training dataset,

wherein the trained supervised machine learning algorithm is used to detect policy violating applications.

14. The method according to claim 13 wherein the obtaining bad applications from the application store comprises:

retrieving applications from the application store;

determining applications that violate policies of the application store, and

classifying applications that violate policies of the application store as bad applications.

15. The method according to claim 14 wherein the retrieving the applications from the application store comprises:

obtaining a listing of applications from an external source; and

retrieving applications contained in the listing of applications from the application store.

16. The method according to any one of claims 14 or 15 wherein the determining the applications that violate the policies of the application store comprises:

determining, after a period of time, if the retrieved applications are contained in the application store,

wherein retrieved applications that are not contained in the application store after the period of time are classified as bad applications.

17. The method according to any one of claims 13-16 wherein the generating the feature table comprises:

compiling the blacklist based on a set of common terms, Intellectual Property (IP) violating terms, third-party libraries having aggressive advertisement behaviour and identities of undesirable computer servers;

compiling the permission list based on permissions that access secure device data and device access control data;

compiling the API call list based on a set of predetermined API calls and obfuscated API calls;

compiling the metadata list based on a set of predetermined metadata,

whereby the blacklist, the permission list, the API call list and the metadata list are each assigned a column in the feature table and each bad application is assigned a row in the feature table.

18. The method according to any one of claims 13-17 wherein the obtaining the parameters for bad applications using the feature table comprises:

for each bad application contained in the feature table,

setting a blacklist parameter using identified extracted features;

setting a permission parameter using determined extracted features;

setting an API call parameter using established extracted features; and

setting a metadata parameter using discerned extracted features.

19. The method according to claim 18 wherein the setting the blacklist parameter using the identified extracted features comprises:

extracting features from the bad application;

identifying terms and libraries from the extracted features that match terms and libraries contained within the blacklist;

identifying package files from the extracted features that interact with servers contained in the blacklist; and

setting the blacklist parameter using the identified extracted features.

20. The method according to claim 18 or 19 wherein the setting the permission parameter using the determined extracted features comprises:

extracting features from the bad application;

determining permissions from the extracted features that match permissions contained within the permission list; and

setting the permission parameter using the determined extracted features.

21. The method according to any one of claims 18-19 wherein the setting the API call parameter using the established extracted features comprises:

extracting features from the bad application;

establishing API calls from the extracted features that match API calls contained within the API call list; and

setting the API call parameter using the established API calls.

22. The method according to any one of claims 18-21 wherein the setting the metadata parameter using the discerned extracted features comprises:

extracting features from the bad application;

discerning metadata from the extracted features that comply with metadata queries contained within the metadata list; and

setting the metadata parameter using the discerned metadata.

23. The method according to any one of claims 18 to 22 wherein the extracted features from the bad applications comprise brand names, IP terms, network activities, third-party libraries, permissions, API calls or metadata.

24. The method according to any one of claim 13-23 wherein the supervised machine learning algorithm comprises algorithms based on regression models, support vector machine models, K-nearest neighbour models, base classifier models, simple decision tree models or decision tree models.