Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

Appropriate Conversion of Machine Learning Data

Dimitar Todorov[1], *Milena* Karova[1]

1 - Technical University of Varna, Department of Computer Science and Engineering, 9010, 1 Studentska Street, Varna, Bulgaria

Coresponding author contact: dimgeto@abv.bg

Abstract. Data is an important part of computer technology and, as such, explains the strong dependence of machine learning algorithms on it. The operation of any corresponding algorithm is directly dependent on the type of data and the proper data representation increases the productivity of these algorithms. Advanced in the present article is an algorithm for data pre-processing in a form that is most suitable for machine learning algorithms, with cryptographic secret keys being used as input data. The experimental results were satisfactory, and with the utilization of secret keys with significant differences, the recognition obtained is about 100%.

Keywords: homogeneous environment, machine learning, data, k Nearest Neighbours, Support-Vector Machines, secret key

1 INTRODUCTION

The selection of an appropriate type of machine learning (similarly an appropriate machine learning algorithm), according to the data it will work with, is a key role for its work. For example, one type of algorithm is suitable for working with product pricing or process optimization, another type is suitable for image classification, customer or product segmentation, and so on.

The dependence of machine learning algorithms on data is so great as to be rightfully defined as genetic. Clearly, this is determined by the characteristics of an algorithm itself.

Almost anything can be turned into data. Building an in-depth understanding of different types of data is a crucial prerequisite for performing exploratory data analysis and engineering the main functions for the machine learning models. For example, it is necessary to convert the data types of certain variables in order to make an appropriate choice for visual coding when visualizing data.

The main goal of the experimental setup is the presentation and implementation of an algorithm for appropriate transformation and presentation of data suitable for use by machine learning algorithms.

2 DATA TYPES

Most data can be categorized into four main types in terms of machine learning:

- Numeric data - These are any data where the data points are exact numbers. From the point of view of statistics, they can be called quantitative data. These data, for example, are important, such as establishing the price of housing in a city or their number, or the number of houses being sold in the last year. Numerical data can be characterized as continuous or discrete data. Continuous data can take any value within a certain range (e.g., data on height, weight, temperature, etc.), while discrete data have different values (e.g., data on units sold, number of students, number of languages spoken, etc.). n.). They are not arranged in time, but in arrays of numbers to be collected for a specific purpose.
- Categorized data - They represent characteristics such as a person's nationality, education, hometown, etc., and can accept numeric values. For example, the number 1 can be used to denote red and 2 to denote blue. These numbers, therefore, have no mathematical significance. They cannot add up or take the average. In the context of classification, categorized data would be the class label.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

- Time interval data - These are a sequence of numbers collected at regular intervals over a period of time and refer to critical data in certain areas of finance, for example. They have a time value attached to them that would be a date or time stamp through which trends in time can be sought out. For example, you can measure the average number of sales of a product across the years. Unlike numeric data, which has no time order, the time interval data has some default order as they have the first collected point and the last collected data point.
- Text – The text data is basically just words, and in some cases, they are just single characters.

2.1 Selection of appropriate data

For the purposes of the experimental setup, secret keys from the synchronous encryption algorithms AES, DES, TripleDES and RC2 are used as input data (Antonov, 2000; Sarkar, 2008). The aim is to achieve recognition of the type of encryption algorithm of a secret key from machine learning algorithms. This is of paramount importance in terms of the choice of both the machine learning algorithms used and the type of data to be used for their training. The primary data is the secret keys themselves, which are a series of symbols of different lengths. This determines the text type of data used to achieve the goals of the experimental setup, as each key is a string of separate characters. Figure 1 shows exemplary AES secret keys in hexadecimal form. Working with a large number of characters is also not an easy task for any system. The more they are, the greater the number of references to the knowledge gained so far. Even from a human point of view, it is easier to classify individuals through fewer distinctive features. For example, it is much easier to regroup a group of ten people to short and tall than to regroup them according to their intellectual abilities. In the first it is enough to take into account that the average height for the human individual is 1.7 meters and all those above it can be defined as tall, and below it - short. In addition, such a regrouping can be done without keeping any statistics since it can be determined visually. The second regrouping will require more in-depth research, testing, etc. Similar approach has also been taken to machines as well with most of the criteria leading to more references and comparisons. In the context of computer systems, this will literally result in dozens of CPU accesses and more CPU cycles. The fact is that we are working with data that are relatively large, the location of which is inevitably on disk space in the form of files. The processor's access to the data thus located is always associated with additional time.

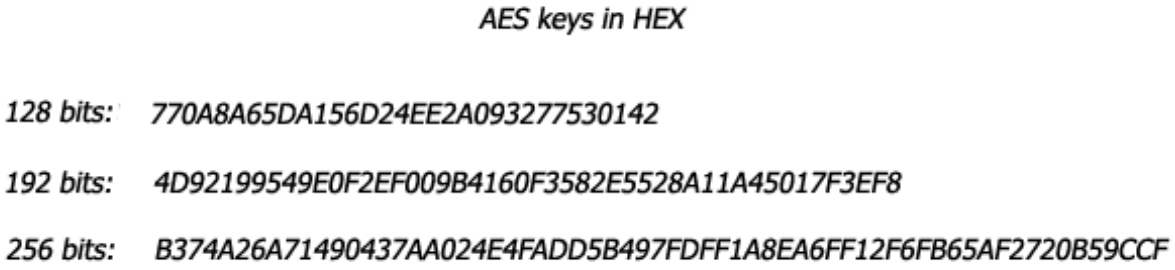

Fig. 1. Sample AES secret keys

Determined, from what has been discussed so far, are the following criteria for the selection of appropriate data:

- To use as few characters as possible;
- To enable easy and bidirectional conversion to the real ones - the string of secret key characters;
- To have universality;
- To allow hardware implementation;
- Ensure they are placed in a simplified form;

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

- Files should be as simple as possible and as small as possible;
- The file format should be able to migrate between different types of operating systems;

In order to fulfill the main task and goal of the experimental setup and the criteria listed above, the binary type of data representation in a simple string of characters placed in a plain text file was chosen. The binary type guarantees the use of only two types of symbols 1 and 0 (Fig. 2). Binary representation is embedded in current computer systems. The converting from and to binary real data - secret keys - is a "familiar" and commonly used operation in a computer system. In practice, the even key generation can be done directly in binary form. There is hardly a more universal way of presenting data from software and hardware. Implementations of AES instructions are built into modern public processors. The possibility of this data type is determined for future conversions. It is possible to place them in any "ordinary" file, regardless of the code table. It should be noted that each character code table can be interpreted in binary form. A text file is chosen for data storage because it is easy to interpret and process in the middle of any operating system. It is easy to create, edit and transfer.

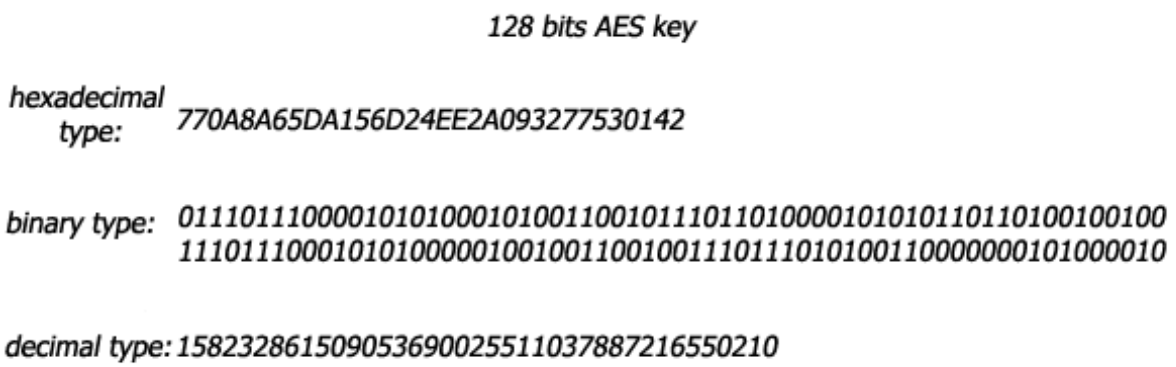

Fig. 2. AES key presented in different form

3 HOMOGENEOUS ENVIRONMENT

Once the data type is determined, it is time to determine the environment in which they will exist during the implementation of the algorithm. It must be consistent with the data, in symbiosis with it and create sufficient contrast to the representation of the different data units. This contrast is of crucial importance for the learning machine learning algorithms. The clearly defined individual data unit differences lead to more accurate operation of these algorithms.

The term homogeneous environment is widespread in many fields of science. Everywhere its definition is different, due to the characteristics of the specific field and the purpose of its use. For example, a manufacturer-based definition is common on the Internet as a concept by technology companies and their customers. It is considered that if a user uses software and hardware from one manufacturer (eg. Microsoft, Apple, etc.), they have achieved a homogeneous environment of the technologies being used.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

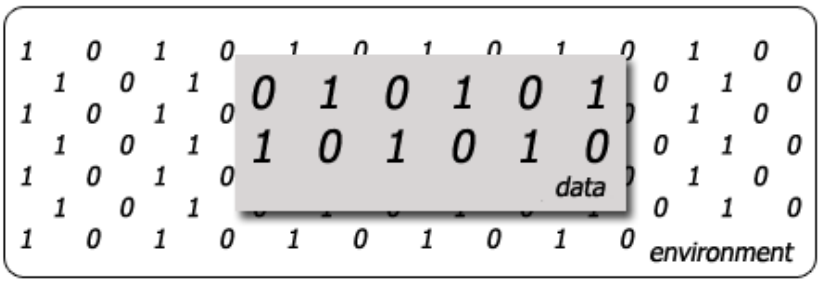

Fig. 3. Homogeneous environment

In order to achieve the goals of the experimental setup and the chosen means and methods, it can be determined that a homogeneous environment is an environment in which the data and their environment are presented with the same type, size and number of symbols of equal total length (Fig.3). The type of symbols according to the bit representation of the data is 1 and 0, and the total length with the middle is selected to be 1024 characters. Each character has a size of 8 bits or 1 byte, and a text file storing data for a key is 1 kilobyte.

4 ALGORITHM FOR PLACING IN A HOMOGENEOUS ENVIRONMENT

The algorithm must be able to work with the data described above and the environment in which it will be placed. In addition, it must achieve its goal of providing an appropriate type of training data and for the operation of machine learning algorithms. In this regard, two main stages in its work can be identified - the transformation of data into an appropriate form and placing them in a homogeneous environment (Fig. 4). After the implementation of these two stages, it is possible to interact with the algorithms for machine learning using the already processed data.

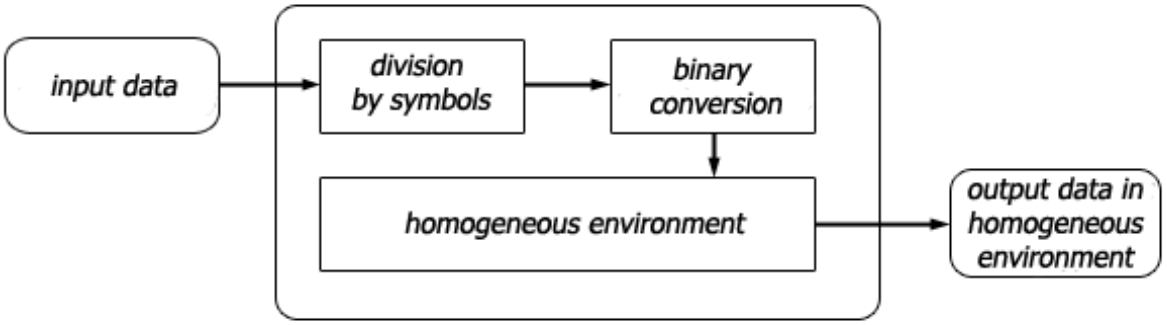

Fig. 4. General view of the algorithm for placing data in a homogeneous environment

4.1 Stage 1: Transformation into an appropriate form

The first step in the operation of the algorithm is to convert the data into an appropriate form. As explained earlier, the species is selected according to the goals and objectives to be achieved and solved by the system. For the purposes of the present dissertation, binary data are selected. It guarantees the use of only two types of symbols 1 and 0, which on the one hand determines the ease of placing the data in a homogeneous environment, and on the other hand favors the use of bipolar classification of controlled machine learning. The two machine learning algorithms (kNN, SVM) hereto applied belong to the controlled machine learning. The use of bipolar classification is a prerequisite for achieving better end results.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

Binary representation is embedded in current computer systems. Almost every programming language provides such an opportunity through ready-made solutions and libraries. The realization of such a transformation is done with the help of several mathematical operations. The cryptographic algorithms used for the purposes of the dissertation use ASCII table symbols to generate their secret keys. Each symbol in the table has its own unique decimal number. It is this number that is converted to binary and gives the binary image of each of the symbols included in the table. The length of each of the characters is 8 bits (1 byte). For example, the symbol "B" (capital letter B) is a decimal number 66. Its conversion into binary form is done by dividing by two, followed by cyclic division of the quotient by two until it becomes equal to 0. The remainder of each operation determines the corresponding bit.

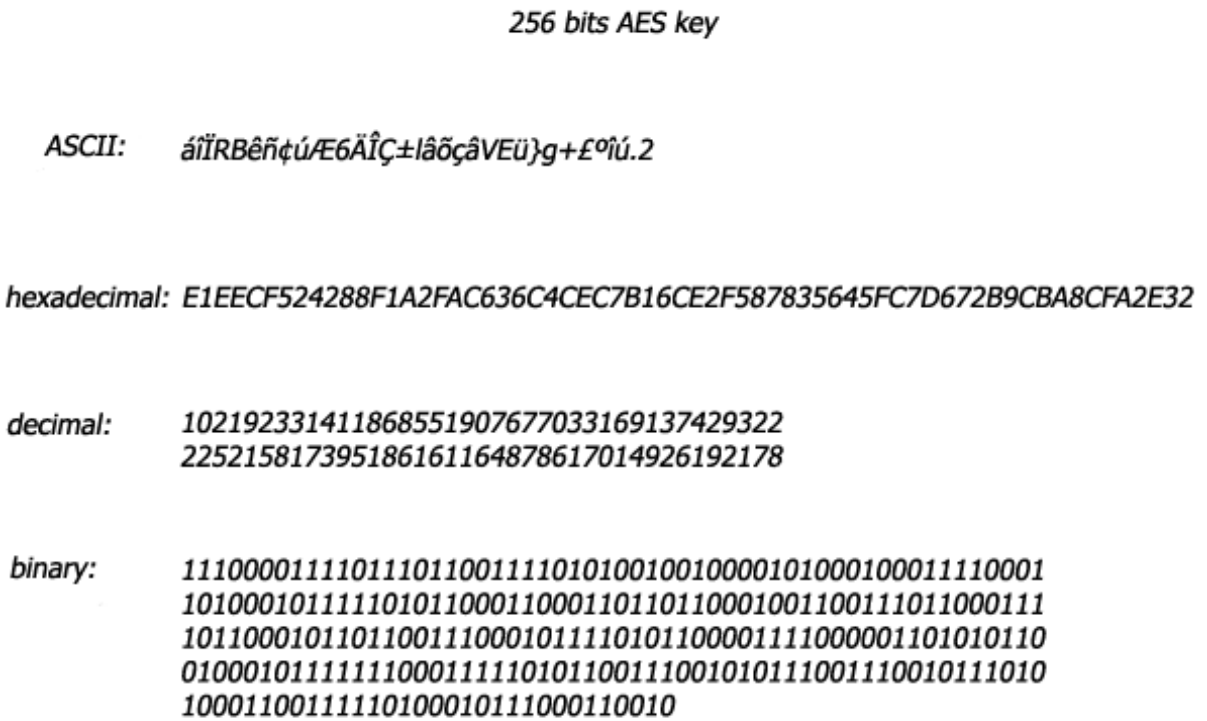

Fig. 5. Differently presented AES key with 256-bit length

Thus, if we take any 256-bit AES secret key after it is converted to binary it will look like the one shown in Fig.5. The key transformed in this way is ready and suitable for placement in a homogeneous environment.

4.2 Stage 2: Placement in a homogeneous environment

The environment where the converted data will be placed is decisive for the operation of the algorithm. It must be consistent with the data and provide sufficient contrast. Something necessary and important for the operation of machine learning algorithms. It has already been defined that a homogeneous environment is an environment in which the data and their environment are represented by the same type, size, and number of symbols of the same total length. It becomes clear, from the above, that the symbols 1 and 0 or only one of the two will constitute the "environment" of the data. From the point of view of contrast, the use of both will lead to its reduction at the same time, and hence, running into more difficulty to distinguish the data. There are two options left - 1 or 0. Both are equivalent from their point of view. The difference between the two is more subjective than practical. Somehow 0 (zero) is naturally perceived as the beginning, the basis, etc. In reality, in electrical engineering, and in electronics, this is literally. For example, for mass logic elements operating with TTL levels, zero is the basis, and low logic levels considered logical 0 to be in the range of 0-0.4 to 0.8V in 5V and 3.3V

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

variants. In another example, zero can be considered as white, and one as black in an art production. Thus, the placed key binary version in an environment supplemented by 0 would look like an imprint (drawing, signature) on a white sheet. In addition, in the preliminary developments of the created software model for the dissertation purposes, a zero is required as a better option for filling in the middle after placing the key binary imprint.

Another factor that a homogeneous environment must take into account is its length. It is important because the data contrast in this environment is again dependent on it. At low lengths there is a danger of great overflow with the possibility of failing to fit even one key in its entire length, which is completely unacceptable. In preliminary developments, a length of 1024 characters or 1024 bytes was required as "successful" because at this length, the results of preliminary developments show that it is optimally minimal to achieve the goals. The length, as a figure of 1024, was chosen to be symmetrical to the presented data, since it is this symmetry that may also allow for a potential hardware implementation of the algorithm.

In reality, each symbol of the secret key is presented in binary form and placed in a homogeneous environment, supplemented by 0 from the left to the required dimension. The complementation on the left is perceived similar to the natural direction of data writing. There is no obstacle to be filled in on the right or on both sides. This is rather a matter of preference in the interpretation of the algorithm, as no contrasting differences between the three options have been observed. The complementation with 0 is realized by means of the coefficient K_{eq}, by which the necessary filling up to the determined dimension is specified.

$$K_{eq} = \frac{L_e}{S_{key}/8} = 8\frac{L_e}{S_{key}} \quad (1)$$

In the formula above, L_e is the assumed (predetermined) length of the homogeneous medium, and S_{key} is the size of the key in bits, which is equalized to bytes by simply dividing by 8.

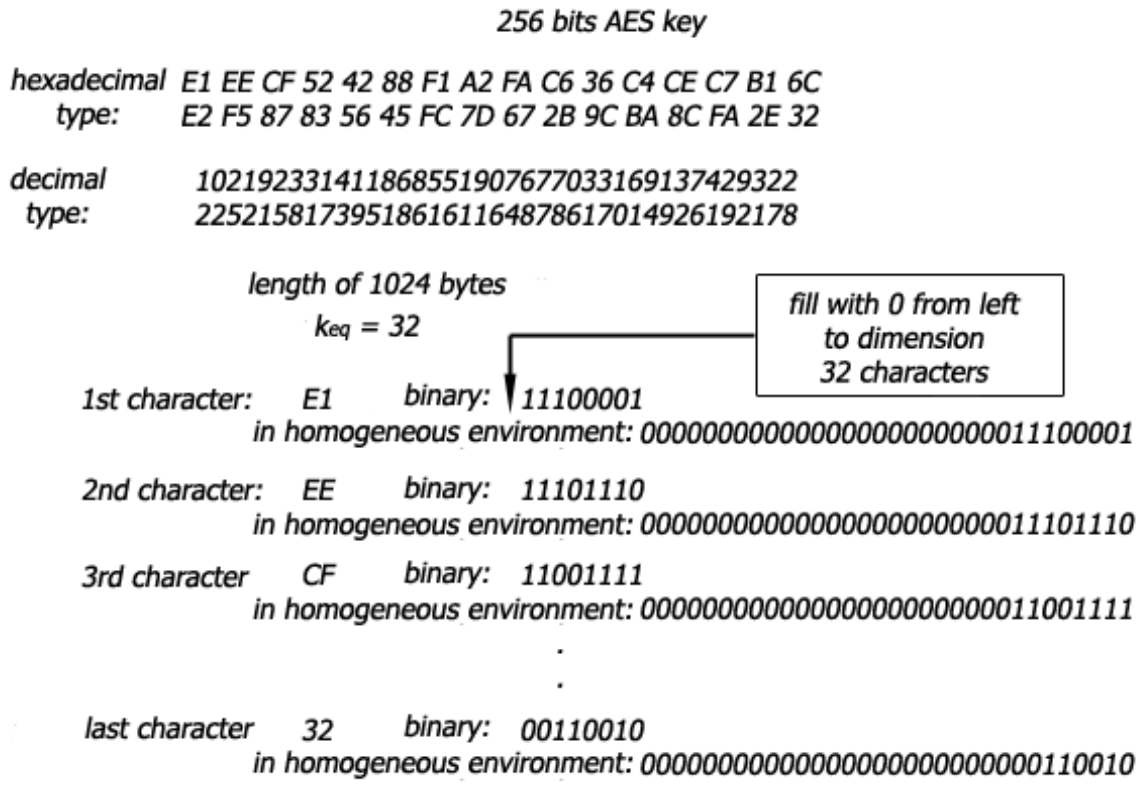

Fig. 6. Fill in the left to a certain dimension

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

For example, at 256-bit AES the key $S_{key} = 256$, the size of the environment is $L_e = 1024$, and the coefficient $K_{eq} = 32$. The transformation and placement of each symbol in a homogeneous environment is shown in Fig.6. After converting each number to binary, it is supplemented by 0 from the left to dimension 32. Or as a result in the case of the sample AES key, 32 parts of 32 elements in total dimension of 1024 are obtained.

It is interesting to note that the filling and placement in the homogeneous medium by means of the coefficient K_{eq} determines a proportional relationship between the size of the secret key and the number and size of the binary fingerprints of the symbols of the secret key.

- at 64 bits - 8 portions of 128 bytes (characters);
- at 128 bits - 16 portions of 64 bytes;
- at 256 bits - 32 portions of 32 bytes;
- at 512 bits - 64 portions of 16 bytes;

The length of the homogeneous medium thus selected determines the use of a maximum key length of 512 bits. If you need a larger key, you need to change the length of the middle and move to a dimension of 2048 bytes, for example. For example, a need will arise when using a 1024-bit key. However, it will be necessary to drop the keys from the DES algorithm. The reason is that they are 64 bit and their fingerprint will differ too easily compared to that of a 1024 bit key.

4.3 Interaction of the algorithm for placing data in a homogeneous environment with machine learning algorithms

The proposed algorithm is designed and developed to present the data in a form suitable and convenient for use by machine learning algorithms to achieve the objectives of the dissertation. This main purpose determines the data as a means of interaction between algorithms. Constructed in the way described so far, the algorithm, in addition to processing and submitting data to machine learning algorithms, can also be used to create basic or so training data (Fig.7).

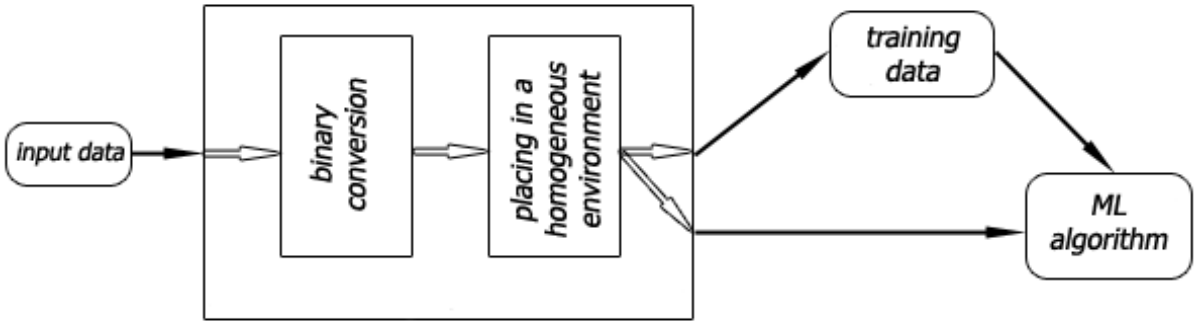

Fig. 7. Scheme of interaction

The operation of the proposed algorithm for placing data in a homogeneous environment is optimized for the actual data with which the machine learning algorithms will work. The selection of the type, type and size of data are the basis of its design and implementation.

4.4 The complexity of the proposed algorithm

To determine the complexity of the proposed algorithm through the use of asymptotic notation based on the functions that characterize the execution time of the algorithms. It can also be applied to functions that characterize some other aspect of the algorithms (for example, the amount of space they use), or even to functions that have nothing to do with algorithms (Nakov, 2015). This is the so-called

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

Big O notation, which is a mathematical notation that describes the limiting behavior of a function when the argument tends to a certain value or infinity. For the use of this notation and for it to be applicable to the execution time of the algorithm, it is necessary to determine which runtime is meant. In some cases, the attention is directed to the worst-case scenario. Very often, however, the goal is to characterize the working time, whatever the input. In other words, it is often necessary to make a general declaration that covers all options, not just the worst case. According to the proposed theory (Nakov, 2015) when compared to the algorithm for placement in a homogeneous medium it is $O\ (n^2)$. Assuming that cyclic action n (the most labor-intensive) is required to traverse the data when converting the data to binary and placing another cyclic action n in the homogeneous medium, the mathematical reference is:

$$2n = O(n\text{^}2) \tag{2}$$

5 APPLICATION OF THE EXPERIMENTAL STAGING

In order to achieve the goal of the experimental staging and to conduct experiments, it is necessary to create a software product. With its help, the broad set of those mentioned in the dissertation should be checked and possibly supported. It must be realized through the conduct of various experimental productions, with certain criteria and parameters.

The requirements for the software product are consistent with all the factors that should be taken into account, namely:

- The main goal of the dissertation.
- Achieving the defined tasks.
- Conducting experiments.

The construction and development of the software product are in accordance with the goals and the formulated tasks for implementation. In this regard, several main nodes are outlined for the implementation of the required functionality of the application (Fig. 8). These are:

- Software implementation of secret key generation for each of the encryption algorithms used to generate secret keys, such as input data.
- Program implementation of data processing in the corresponding selected type and placing them in a homogeneous environment.
- Implementation of modules for each of the two machine learning algorithms.
- Module for the generation and labeling of the training data in line with the size of the homogeneous environment.
- Module for conducting single experiments and complex ones with the necessary consideration as per the defined criteria and parameters.
- Well-informed and functional graphical user interface to enable the use of the product for educational purposes.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

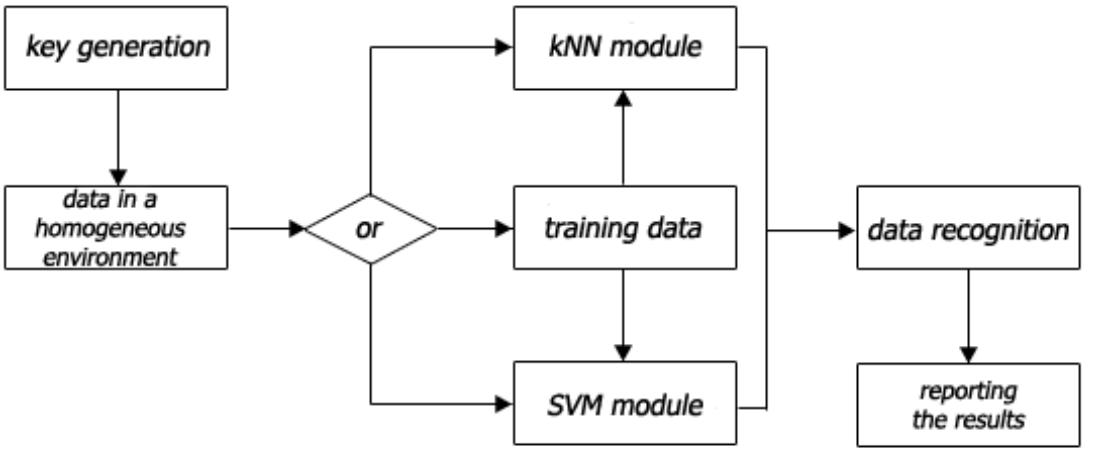

Fig. 8. Block diagram of the application

Two of the most well-known and widely used controlled machine learning algorithms have been used - kNN (k-Nearest Neighbors) and SVM (Support Vector Machines) (Katz, 2015; Marsland, 2015). It is through them that the two recognition modules, which will use the pre-processed data through the algorithm for placing data in a homogeneous environment, are implemented.

5.1 kNN module

The first module of the block diagram implements the algorithm of the nearest neighbor - kNN (k-Nearest Neighbors) (Harrington, 2012; Smola, 2008). In short, the kNN algorithm consists of three main steps:

- Calculating the distance between two points;
- Finding the nearest neighbors based on the calculated distances;
- Choosing the respective class based on the list of nearest neighbors;

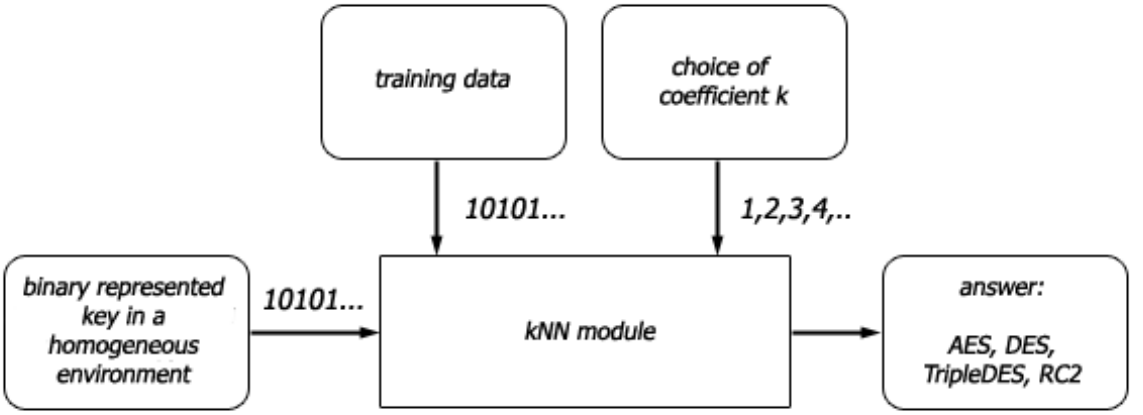

Fig. 9. Block diagram of the kNN module

Figure 9 shows the implementation and operation of the module using the kNN algorithm.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

5.2 SVM module

The other module implements the SVM (Support Vector Machines) algorithm (Flach, 2012; Géron, 2019; Shalev-Shwartz, 2014). It consists of two stages (Fig.10). In the first, a vector machine is created and trained with the available basic data. In the case of known keys appropriately transformed and placed in a homogeneous environment. The second stage realizes the active, real work of the trained vector machine for the unknown key classification.

The algorithm builds a classifier, which is a dividing line between two data classes. The first class is one of the possible classes (in the first iteration - AES key, in the next iterations - DES key, TripleDES key, RC2 key). The second class groups all the other classes that have not been checked until the current iteration.

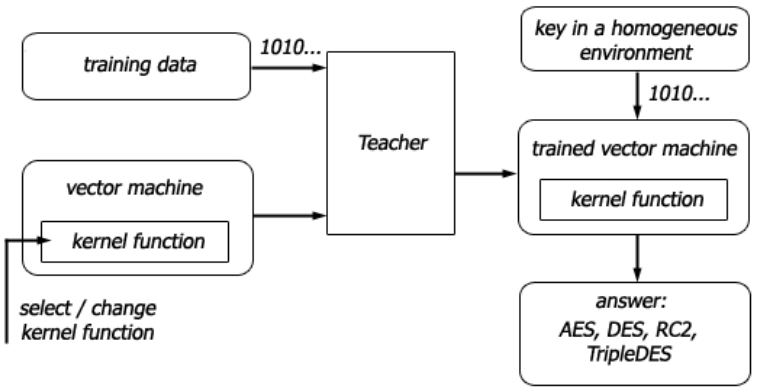

Fig. 10. Block diagram of the SVM module

6 EXPERIMENTAL RESULTS

The application and the presented modules were developed in the C # software environment, using the machine learning platform Accord.Net. It was realized for the purposes of the experimental production and was installed and tested on the following computer configuration:

- Processor Intel Core i5 – 11400H 2.7 / 4.5 GHz;
- 16 GB DDR4 3200MHz;
- Windows 10 Pro 64-bit (x64);
- .NET Framework 4.5;
- Accord.Net Framework 3.8;

6.1 Consumption of system resources

The application runs normally. During the operation, it takes up from 16 to 60 MB in standby mode and reaches about 180 MB of the entire RAM space. During the data classification, the application engages the processor up to about 80%. The use of significant system time is observed in the implementation of the module implemented using the algorithm using the method of support vectors (SVM, Support Vector Mashine). This phenomenon is completely justified and expected in the design of the application for the simple reason that the algorithm uses relatively complex mathematical functions.

6.2 Results of the conducted experiments

When specifying the input parameters, different cryptographic algorithms are selected - with different key lengths of 64, 128 and 256 bits. The aim is to achieve a variety of input data. The results are best at both extremes at 64 and 256, because their footprint is more contrasting than the other two (128 bits) placed in a homogeneous environment. In TripleDES and RC2, despite the inclusion of the given

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

sample data, and aside from the competition of the given fingerprint with another similar one of the same length, the result in both modules (kNN and SVM) is nevertheless close to about 55 - 65%.

Attempts were made with a total of 400 keys of 100 for each encryption algorithm, recognized by each of the two machine learning algorithms - kNN and SVM, using basic data from 4000 known examples obtained by the algorithm for placing data in a homogeneous environment. The results for the module with kNN are shown in Tables 1, 2 and 3. The execution time of one recognition is less than 1 s (on average about 249 ms), and the execution time of all 400 examples is about 10 minutes. The percentage results are very good at 256 and 64 bit key size, and the second is even 100 %. For 128 bit (TripleDES and RC2) the results are rather good, but definitely weaker compared to AES and DES.

Table 1. Test result with kNN in a homogeneous environment with a length of 1024

homogeneous environment						1024
Algorithm	Key size	Correctly recognized	Incorrectly recognized	%	Load known keys, s	Recognition, s
AES	256	80	20	80,00	0,0005	0,198
DES	64	100	0	100,00	0,0005	0,199
TripleDES	128	53	47	53,00	0,0005	0,217
RC2	128	54	46	54,00	0,0005	0,218
					Total time, min	
Total (average)		287	113	71,75	10,120	

Table 2. Test result with kNN in a homogeneous environment with a length of 1536

homogeneous environment						1536
Algorithm	Key size	Correctly recognized	Incorrectly recognized	%	Load known keys, s	Recognition, s
AES	256	75	25	75,00	0,0005	0,265
DES	64	100	0	100,00	0,0005	0,241
TripleDES	128	70	30	70,00	0,0005	0,242
RC2	128	55	45	55,00	0,0005	0,253
					Total time, min	
Total (average)		300	100	75,00	9,871	

Table 3. Test result with kNN in a homogeneous environment with a length of 2048

homogeneous environment						2048
Algorithm	Key size	Correctly recognized	Incorrectly recognized	%	Load known keys, s	Recognition, s
AES	256	74	26	74,00	0,0005	0,291
DES	64	100	0	100,00	0,0005	0,275
TripleDES	128	62	38	62,00	0,0005	0,301

Technical University of Varna
Annual Journal
ISSN 2603-316X (Online)
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31

RC2	128	63	37	63,00	0,0005	0,274
					Total time, min	
Total (average)		299	101	74,75	9,871	

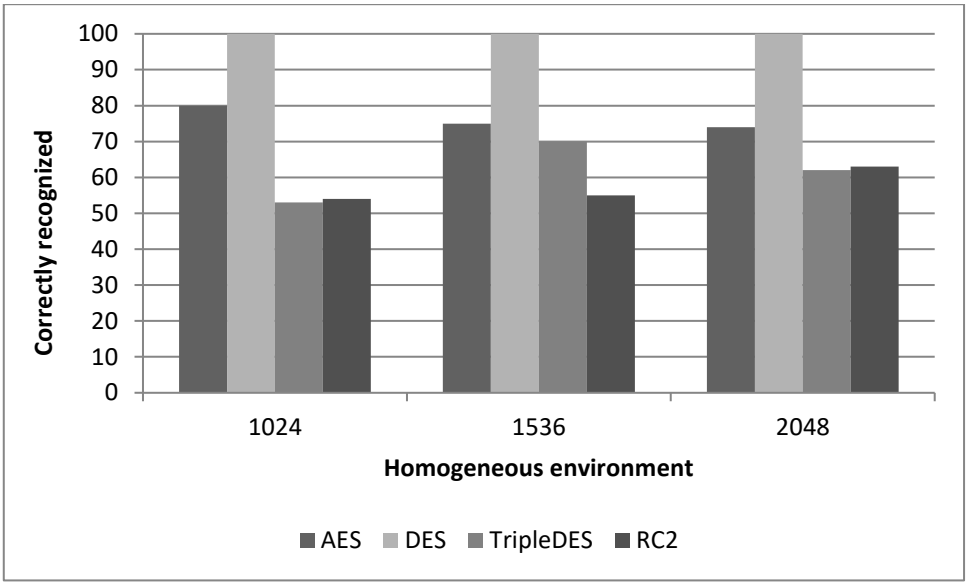

Fig. 11. Graph of correct recognition with kNN in different environments

Figure 11 shows the dependence of the correct recognition of kNN data on the size of the homogeneous medium.

The results for the SVM module are shown in Tables 4, 5 and 6. The execution time of one recognition is on average about 10 s, and the execution time of all 400 examples is on average about 50 minutes. There is a significant difference in the recognition times with increasing size of the homogeneous medium. At size 1024 the time for one recognition is about 6-7 s, and at 2048 it is almost double - about 13.45 s. The percentage results are excellent for AES and DES, and both are 100%. For TripleDES and RC2, the results are impressive, though contrastingly weaker than AES and DES. There is also a downward trend in the results with increasing size of the environment.

Table 4. Test result with SVM in a homogeneous environment with a length of 1024

homogeneous environment						1024
Algorithm	Key size	Correctly recognized	Incorrectly recognized	%	Load known keys, s	Recognition, s
AES	256	100	0	100,00	0,005	6,831
DES	64	100	0	100,00	0,005	6,902
TripleDES	128	62	38	62,00	0,005	7,729
RC2	128	69	31	69,00	0,005	6,911
					Total time, min	
Total (average)		331	69	82,75	40,56	

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

Table 5. Test result with SVM in a homogeneous environment with a length of 1536

homogeneous environment						1536
Algorithm	Key size	Correctly recognized	Incorrectly recognized	%	Load known keys, s	Recognition, s
AES	256	100	0	100,00	0,005	10,291
DES	64	100	0	100,00	0,005	10,156
TripleDES	128	63	37	63,00	0,005	10,283
RC2	128	52	48	52,00	0,005	6,911
					Total time, min	
Total (average)		315	85	78,75	62,099	

Table 6. Test result with SVM in a homogeneous environment with a length of 2048

homogeneous environment						2048
Algorithm	Key size	Correctly recognized	Incorrectly recognized	%	Load known keys, s	Recognition, s
AES	256	100	0	100,00	0,005	13,46
DES	64	100	0	100,00	0,005	13,443
TripleDES	128	52	48	52,00	0,005	13,711
RC2	128	52	48	52,00	0,005	13,308
					Total time, min	
Total (average)		304	96	76,00	62,099	

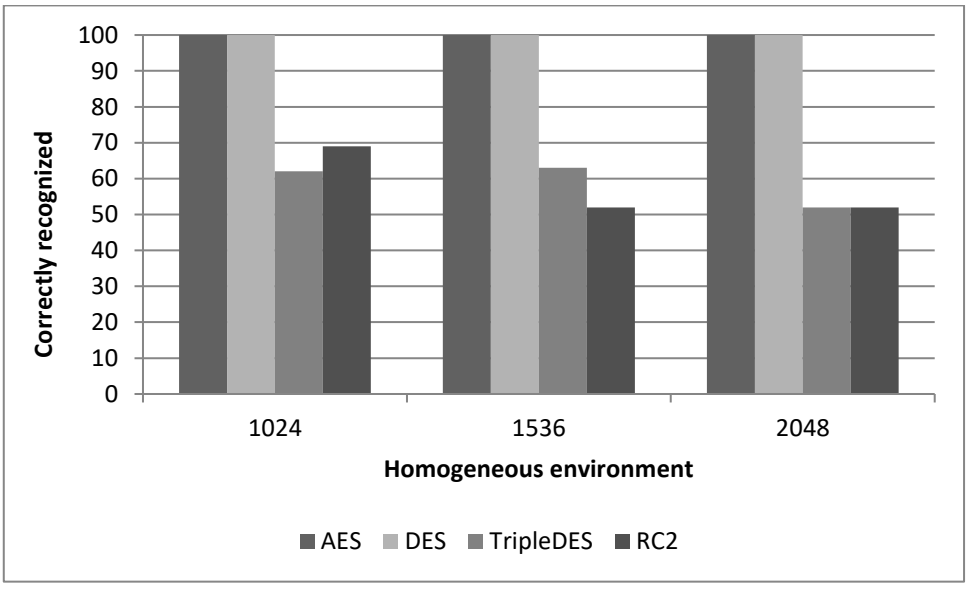

Fig. 12. Graph of correct recognition with SVM in different environments

Figure 12 shows the dependence of the correct recognition of SVM data on the size of the homogeneous medium.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol6.iss2.262
Vol. 6 Issue 2 (2022)
Published: 2022-12-31
ISSN 2603-316X (Online)

7 Discussion

In summary, there are two topics that stand out. The first, with SVM, the overall percentage results are better by about 6-8%. The second, the time required to perform the experiment with all 400 examples in SVM is on average about 50 minutes versus 10 minutes in this ratio at kNN. In addition, with increasing size of the homogeneous medium, there is no increase in contrast and hence the number of correctly recognized examples. The size must be determined in accordance with the input data and specifically with their size so as not to overflow with insufficient size of the homogeneous medium. The excess and the size would lead to data dilution.

Considered at the next stage should be the results of increased dimensionality of the homogeneous environment in which the keys are placed and the number of known examples or so called base data at the expense of hardware load. The introduction of this encryption type and the decryption method in the sender-receiver communication environment is upcoming and its stability in the communication environment will be studied.

8 Conclusions

When the input parameters are specified, the different cryptographic algorithms are deliberately selected in this form - with different key lengths of 64, 128 and 256 bits. As expected, the results are best in both extremes at 64 and 256, because their footprint is more contrasting than the other two (128 bits) placed in a homogeneous environment. Although the inclusion of the given sample data in TripleDES and RC2 in addition to the competition of the given fingerprint with another similar one of the same length, the result in both modules (kNN and SVM) is close to 50%.

The conducted experiments prove the significant effect of the algorithm for placing data in a homogeneous environment on the work of the two machine learning algorithms. This influence is an important part of achieving good data recognition results. The size of the homogeneous medium should be consistent with the input data. Too large a medium does not necessarily increase the contrast of the recognition data. The algorithm can definitely be used to properly present data on machine learning algorithms.

References

Antonov, P., Malchev, S. (2000). Cryptography in computer communications (pp. 70-164). Varna, Technical University of Varna

Harrington, P., (2012). Machine learning in action (pp.3-149). Shelter Island, USA, Manning Publications Co

Smola, A., Vishwanathan, S.V.N. (2008). Introduction to machine learning (pp.20-32). United Kingdom , Cambridge University Press

Sarkar A., Chatterjee, Chakraborty, M. (2021). Role of Cryptography in Network Security, The "Essence" of Network Security: An End-to-End Panorama (pp 103-143). USA, SpringerLink

Katz, J., Lindell Y. (2015). Introduction to Modern Cryptography (pp). USA ,CRC Press

Flach, P. (2012). Mashine Learning. The Art and Science of Algorithms that Make Sense of Data (pp). UK, Cambridge University Press

Nakov, P., Dobrikov, P. (2015). Programming = ++ Algorithms; Fifth edition (pp). Bulgaria, Software University

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (pp). USA, O'Reilly

Marsland, S. (2015). Machine Learning. An Algorithmic Perspective. Second Edition (pp). USA, CRC

Shalev-Shwartz, S., Ben-David, S. (2014). Understanding Machine Learning. From Theory to Algorithms (pp), USA, Cambridge University Press