Б Ъ Л Г А Р С К А А К А Д Е М И Я Н А Н А У К И Т Е

ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

Иван Иванов Благоев

МЕТОДИ И СРЕДСТВА ЗА АНАЛИЗ НА ДАННИ В ИНФОРМАЦИОННИ СИСТЕМИ С ИЗПОЛЗВАНЕ НА ВРЕМЕВИ РЕДОВЕ

ДИСЕРТАЦИЯ

за придобиване на образователната и научна степен „доктор“

по докторска програма „Информатика“

професионално направление 4.6. “Информатика и компютърни науки“

Научен ръководител: доц. д-р Татяна Атанасова

София, 2021 г.

Списък на използвани съкращения и означения

ИТ - Информационни Технологии

AI - Artificial Intelligence - Изкуствен интелект

MA - Moving Average - Пълзяща средна линия

SMA – Simple Moving Average

EMA – Exponential Moving Average

RNG - Random Number Generator

PRNG - Pseudo-Random Number Generator

OTP - One Time Password

CSV - Comma-Separated Values

RSA - Rivest Rivest-Adi Shamir-Leonard Adleman - асиметричен алгоритъм за криптиране

GCD - Greatest Common Divisor - най-голям общ делител

MLP – Multilayer Perceptron - Многослоен перцептрон

VPS - Virtual Private Server - виртуален частен сървър

PHP – “"Personal Home Page”- скриптов език върху сървърната страна

SSL - Secure Sockets Layer- криптографски протокол за връзка клиент-сървър

TLS - Transport Layer Security - криптографски протокол с асиметрична криптография

IP - Internet Protocol - протокол за комуникация

DNS - Domain Name System - система за имена на домейните

HTTP - Hypertext Transfer Protocol -протокол за пренос на хипертекст

Съдържание:

Увод

Напредъка в технологиите е толкова очевиден, че може само да се спомене, без нужда от фактологично описание, за да се доказва. В това отношение, значителна разлика от последно време е силно експанзиращата дигитална трансформация. Поради COVID-19 заплахата за човешкото здраве, скоростта на навлизане на технологиите в нашият живот силно се ускори. Което води до тотална промяна в множество дейности, а в следващите години ще се забелязва още по-силно, когато човечеството се трансформира и адаптира към този нов начин на живот. Много анализи сочат, че различни професии ще изчезнат и ще се създадат нови такива за човешката дейност. Но освен за момента в сферата на анализите, явно тази тенденция е толкова очевидна, че може да се види началото на не малко процеси за преквалификация на кадри от различни сектори.

Всичкото споменато до тук, води със себе си и до много напълно нови за науката и неизследвани до сега процеси. Събирането и обработката на големи данни ще се разшири и с проникването в новите процеси. Нуждата от изследване и нови открития ще е решаваща за развитието на науката и технологиите в следващите години. За това нуждата от обработка на големи данни и изследванията с времеви редове е изключително важна и ще бъде основен инструмент за изследванията и развитието на науката и технологиите в бъдеще.

Дигиталните отпечатъци, резултат от човешка и природна активност, могат ефективно да бъдат използвани за измерване на голямо разнообразие от явления. Времевите редове, описващи тези явления, ще са предмет на анализ от множество подходи, методи и решения. Причината за това е, че наситеността на събития и процеси в тези данни са твърде дълбоки и науката търси все по-ефективни методи, за да анализира описваните процеси. Това ще позволи осъвършенстването в управлението на множество процеси и в предсказването на много явления.

Като основни характеристики на компонентите на времевите редове са:

1. Тенденциите: вариации, които се движат нагоре или надолу по разумно предсказуем модел. Това е движение към относително по-високи или по-ниски стойности за дълъг период от време. Действието се появява за известно време и след това изчезва.

2. Цикличност: този модел съществува, когато данните показват нарастване и спадане, които не са с фиксиран период. Тези вариации съответстват на множество явления от природни до човешка активност.
3. Сезонност: сезонността може да се дефинира като вариации, които се повтарят за определен период от време, като ден, седмица, месец, сезон и т.н., Сезонността се характеризира винаги с фиксиран и известен период.
4. Случайни вариации: тези вариации възникват поради внезапни причини, наречени остатъчни вариации поради нестабилни колебания в данните, които е трудно да се предвидят. Този тип е често предмет на най-много изследвания и не попада в нито една от горните три класификации.

Други явления които се засичат с времевите редове са стационарност и автокорелация. Стационарността е един от важните фактори, които трябва да се имат предвид при работа с времеви редове. Казва се, че времевия ред е стационарен, ако няма промяна в свойствата му през определен период от време. Това предполага, че вземането на последователни проби от данни с еднакви размери трябва да има еднакви ковариянси и идентични разпределения. Автокорелацията може да се дефинира като корелация на времеви редици със закъснение във времето. Това показва, дали предишните стойности на времевия ред оказват влияние върху настоящите стойности или не.

Следователно, след краткото обхождане на предмета на дейност, компонентите и моделите на анализ на времевите редове, може да се каже, че знаем защо това е много важна област на изследвания в науката за данните. Времевите редове позволяват както описателен, така и прогнозен анализ и ще продължат да играят основна роля в индустрията, бизнеса и секторите за планиране на бъдещи операции.

Настоящият дисертационен труд, чрез изследвания с времеви редове допринася за постигане на по-добри резултати при методи за прогнозиране на финансови инструменти, обработката на големи данни и подобряване на криптографията и кибер сигурността.

Структура на дисертацията

Дисертационният труд е структуриран в четири глави.

В **първа** глава е направен преглед на актуалните теми в областта на науката за данните, особенно когато тези данни се представят като времеви редове. Мотивирана е необходимостта от разработване на нови методи и средства за анализ на данни в информационни системи с използване на времеви редове.

Във **втора** глава са представени разработените методи за изследване и прогнозиране на финансовите времеви редове с използване на различни математически апарати.

В **трета** глава са описани разработените решения за осигуряване на криптографска защита при предоставяне на информационни услуги чрез изследване на генератори на случайни числа, представляващи поредици от времеви редове. Представено е практическото приложение на предложените подходи за обезпечение на киберсигурност. Показани са реалните резултати от проведените тестове, доказващи успешното решаване на поставените задачи.

В **четвърта** глава преодоляването на проблеми при работа с големи масиви от данни и ограничени компютърни ресурси при изследване на времеви редове е направено с разработените софтуерни подходи и със средствата на език за програмиране R.

В **Заключението** е представено резюме на получените резултати от разработката. Определени са насоки за бъдещи изследвания и развитие. Представен е списък с научни публикации по темата и забелязани цитирания.

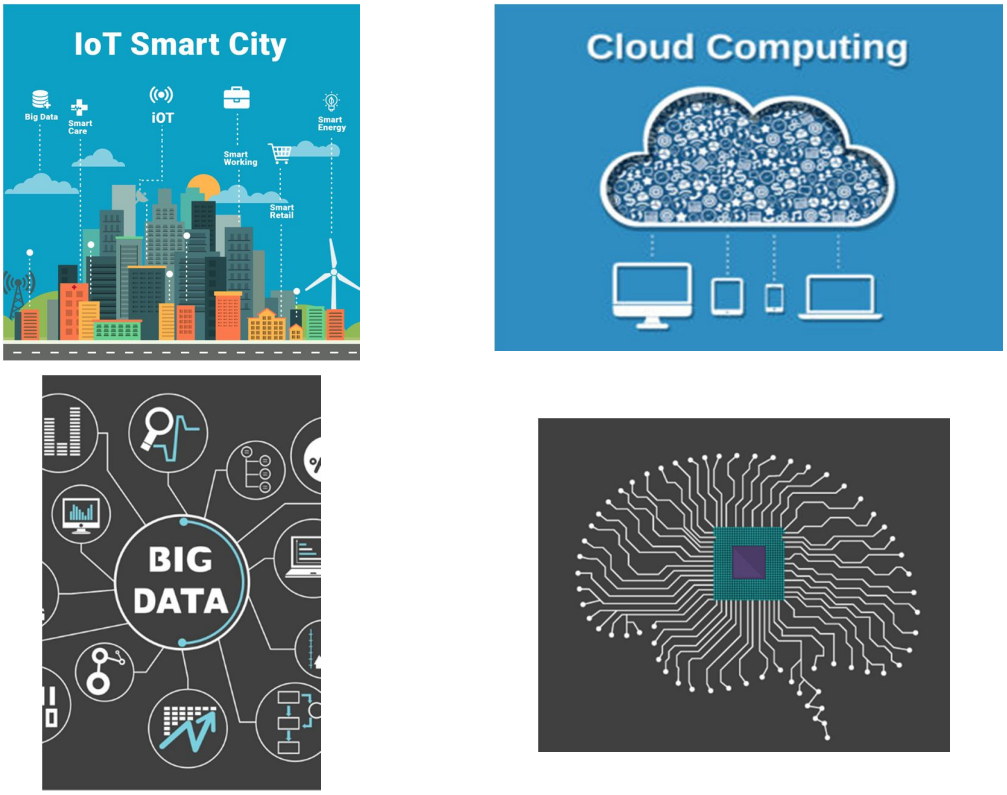
Дисертационният труд съдържа 125 страници, 33 фигури, 1 таблица и 122 литературни източника.

Глава 1. Анализ на състоянието на изследванията.

Значителен тласък за развитието на производителните сили е започнал още през 19 век. благодарение на бързото развитие на естествената наука, която успя да съчетае разнородни знания за света около нас под формата на единна хармонична научна система, която позволява не само да се обяснят много открития, но и да се определят приоритетните области на научните изследвания в дългосрочен план. Това създаде предпоставки за бързото развитие на природните науки, чиито открития започнаха активно да се въвеждат в технологиите и индустрията. Преместваики поглед към днешни дни, научно-техническият прогрес започва да придобива не еволюционен, а революционен характер. Количественото натрупване както на практически умения за използване и усъвършенстване на технически инструменти, така и на научни знания за заобикалящия ги свят прерасна в качествен скок, който направи възможно осигуряването на тясно, непрекъснато ускоряващо се взаимодействие между науката и технологиите.

Ако се погледне в околния свят през окото на технологиите, първото което би впечатлило всеки специалист е колко много данни са това. Данните не са проблем, както в по-далечното минало на технологиите – днес те са навсякъде. Това е страничен ефект от масовата дигитална трансформация и автоматизацията (Wang, 2020), оставяйки цифрова следа от изпълнението на реалния процес. Тези цифрови следи отразяват случващото се в реалния свят и позволяват задълбочен анализ на основните процеси.

Динамичните времеви редове в телекомуникациите, енергетиката, бизнеса идват в резултат на измерване на характеристики от технически, природни, социални, икономически и други системи (Mikalef, 2020), (Ciampi, 2020). Със съвременните тенденции задачите за анализ на времевите редове за откриване на закономерности и тенденции в различни явления и процеси стават все по-важни.



Фиг. 1.1 Съвременните тенденции – IoT Smart City (Righetti , 2018), Big Data (Riahi, 2018) , Cloud Computing (CLOUD, 2019), Artificial Intelligence (Bartneck, 2021)

Анализът на данните, ориентирани към времето, и прогнозирането на бъдещи стойности на времевите редове са сред най-важните проблеми, с които се сблъскват анализаторите в много области, вариращи от финанси и икономика, до управление на производствени операции, анализ на политически и социални политически сесии, както и проучване въздействието на хората и политическите решения, които те вземат, върху околната среда (Montgomery, 2008).

Обработката на големи данни и AI не изостават далеч от нарастващите предизвикателства. При работа с големи масиви от данни възникват много проблеми и ограничения (Mahmud, 2020).

Обяснение за това, какви са големите данни (Big Data) и как те се анализират може да се намери, например, в (Labrinidis, 2012), където са представени и някои казуси, илюстриращи потенциалите и проблемите при анализа на големите данни. Прегледът на различните модели, обхващащи широк спектър от аналитични задачи в областта на науката на данните, е предложен в (Wu, 2014). Открояват се задачите за изследване на големите данни при визуализация на мрежи, децентрализираните и

динамичните оценки (Wang, 2020), прогнозирането на трафика на натоварвания, както и базираните на тензори медицински изображения (Razzak, 2020).

Съвременните изследвания по актуалните теми в областта на науката на данните (Grover, 2020) изтъкват следните важни принципи:

- наборите от данни обикновено са много по-големи от тези, представени в типичните курсове по статистика;
- данните рядко са в правилния формат, за да могат веднага да започнат да се анализират;
- статистическите методи, обработката на данни и представянето на резултатите не трябва да се разглеждат отделно.

Месеци след пандемията на коронавируса, COVID-19 продължава да прекроява ежедневните операции за организации по целия свят. През последните месеци бизнесът прие политики за дистанционна работа, за да намали разпространението на COVID-19. Тази промяна въвежда технически предизвикателства, както и рискове за сигурността на ИТ екипите, които контролират отдалечена работна сила. Covid-19 (Chen, 2020) може да се разглежда като катализатор за нови цифрови възможности.

Всичко това води до нови предизвикателства пред използване и обработка на данни и информация (Staykov, 2019) в различни области на живота и техника, като финансови транзакции (Hasan, 2020), научни пресмятания (Hancock, 2020), киберсигурността и т.н.

1.1 Времеви редове

Времевите редове представляват редици от данни събрани на равни или неравни интервали от време. Представено математически означава, че всяка стойност от времето t може да се съпостави със стойност на определена величина y:

$$y = f(t)$$

Основна характеристика на времевия ред е, че всяка следваща стойност е в зависимост от предходните стойности. Тази зависимост може да бъде, както много сложна, така и относително проста. Често представян пример за времеви ред е средната

дневна температура в рамките на период от време. Но в днешни дни с напредъка на технологиите и големите данни навлизат във всякакви области. В основата на техния анализ и изследване са времевите редове. Чрез тях можем да визуализираме явления, процеси, повтаряемост, сезонност (Ketipov, 2018), колизии, ентропия и колебания на базата на които се правят анализи, прогнози и оптимизации.

Някои примери на различни времеви редове и техните характеристики (Reinert, 2010):

- Поради непрекъснатия характер на химическите производствени процеси, изходните свойства често са положително а*втокорелирани*.
- Времевите редове на производство на храни могат да покажат *линейна тенденция* с постоянен положителен наклон и случайни вариации от година на година.
- Графиката на месечните доставки на напитки разкрива обща нарастваща тенденция, с отчетлив *цикличен модел*, който се повтаря през всяка година.
- Месечната или *сезонната вариация* може да се дължи на въздействието на времето върху търсенето на продукти.
- Бизнес данни като цени на акции и лихвени проценти често показват *нестационарно* поведение; т.е. времевите редове нямат определена средна стойност.
- Графиките от времеви редове могат също да насочат вниманието към появата на *нетипични* събития.

Целите на анализа на времевите редове са:

- описание – извеждане на обобщена статистика и графично представяне;
- анализ и интерпретация – намиране на модел, който описва зависимостта от времето в данните;
- прогнозиране – при дадена извадка от поредицата, се прогнозира следващата стойност или следващите няколко стойности.

Да вземем за пример финансово-технологичният бранш, от последно време решенията предлагани чрез *blockchain – distributed ledger* (Masood, 2018). Общото при този тип технологични решения е, че се събират данни в блокове, в които информацията е защитена от промяна, тези блокове могат да са с определен размер или да са от съответен времеви порядък. Целта е всички действия на потребители на съответното блокчейн решение да се записват в поредицата от блокове по времето на

тяхното удостоверяване. За това блокчейн може да се разглежда, като времеви ред от хронологични събития, в който записаните данни са защитени от промяна и подлежат на съответните за времевите редове анализи.

Анализът на данните, ориентирани към времето, и прогнозирането на бъдещи стойности на времевите редове са сред най-важните проблеми, с които се сблъскват анализаторите в много области, вариращи от финанси и икономика, до управление на производствени операции, анализ на политически и социални сесии, както и проучване въздействието на хората и политическите решения, които те вземат, върху околната среда (Montgomery, 2008).

Нийлс Бор отбелязва: Трудно е да се правят прогнози, особено за бъдещето. (*It is difficult to make predictions, especially about the future. Neils Bohr, Danish physicist*).

Задачите на прогнозирането често се класифицират като краткосрочни, средносрочни и дългосрочни. Прогнозирането на бъдещи събития е от решаващо значение за много видове процеси на планиране и вземане на решения с приложение в области като следните:

- Икономика, финанси и управление на риска;
- Маркетинг и контрол на качеството;
- Метеорология и хидрология;
- Управление на промишлени процеси;
- Демографски данни;
- Медицина и фармакология;
- Моделиране на околната среда;
- Сигурност.

Това са само няколко от многото различни ситуации, при които се изискват прогнози, за да се вземат добри решения. Въпреки широкия спектър от проблемни ситуации, които изискват прогнози, съществуват само два широки типа техники за прогнозиране - качествени методи и количествени методи (Reinert, 2010). Техниките за качествено прогнозиране често имат субективен характер и изискват преценка от страна на експертите. Качествените прогнози често се използват в ситуации, в които има малко или никакви исторически данни, на които да се основава прогнозата. Може

би най-формалната и широко известна техника за качествено прогнозиране е методът Делфи (Dalkey1967).

Техниките за количествено прогнозиране използват исторически данни и модел за прогнозиране. Моделът обобщава тенденции, шаблони и структури в данните и изразява статистическа връзка между предишни и текущи стойности на променливата. Има три основни подхода за генериране на прогнози: базирани на регресия методи, евристични изглаждащи методи и общи модели от времеви редове.

Графиките от времеви редове могат да разкрият такива модели като случайни събития, тенденции, промени в нивата, периоди или цикли, необичайни наблюдения или комбинация от тях.

Прогнозирането на данните от времеви редове е относително сложна задача, тъй като много събития и фактори могат да оказват влияние върху тях.

Понастоящем много методи за прогнозиране, които действат като ефективни инструменти, са широко приети за оценка и анализ на данни от модели на времеви редове (Diggle, 1990; Brockwell, 1991; Smith, 2001; Brockwell, 2002; Shumway, 2006; Golyandina, 2020). От тях, най-често използваният модел е интегриран метод на авторегресия със сезонен компонент (SARIMA - Seasonal ARIMA), който по същество принадлежи към линеен модел. Но на практика при решаване на различни задачи в информационни системи най-често срещаното е, че процесът на генериране на данни е силно нелинеен и прогнозите получени с тези модели, не позволяват да се достигне до точните резултати (Martínez-Acosta, 2020).

1.2. Приложение на времеви редове върху финансови инструменти

Движението на цените на финансовите пазари са прекрасен пример за времеви редове. Техническият анализ, на който се уповават множество от пазарните участници, се основава изцяло на времевите редове и статистиката.

От съществуването на пазари и търговия движенията на цените е от изключителна важност за участниците в тях. В съвременния глобален и силно свързан свят участниците на световния пазар са различни. Производители търсещи клиенти, дистрибутори, инвеститори и спекуланти, всички са обединени от едно - да спечелят.

Дори и дребният потребител се интересува от движенията на цените на потреблението и, ако намери начин, е готов винаги да използва възможности за по-изгодни сделки. Целият този процес е силно мащабен, може да се оприличи на организъм, в който по вени и артерии се придвижват капиталите от участниците на пазара. Но, макар всички участници да следват една и съща цел, не означава, че ще достигат до еднакви резултати.

Елементът на критичност при пазарните ценови движения и разликите в цената за различните участници е напълно различен.

При стабилна икономика, за крайния потребител търсенето да задоволи потребителските си нужди и ако това е на по-изгодна за цена, ще му позволи да спести средства от бюджета, всъщност цените определят неговото потребление.

Печалбата на производители и дистрибутори е силно зависима от резки промени на пазарните цени. Те предпочитат пазарът да е сравнително спокоен и предвидим. За тях е особено важно да се планират добре разходите и приходите напред във времето, за да се гарантира успешно пазарно представяне в условията на голямата конкуренция. За тази група се счита, че е важно да има стабилни очаквания за движението на цените в конкретната дейност в цикъл, като сезон или производствен цикъл. Това позволява да се планират инвестициите в производството и възвращаемостта от направените инвестиции. Неочаквани и продължителни, непредвидени пазарни колебания може да са критични за участниците от този пазарен сегмент и да доведат дори до фалит. За ярък пример може да се вземат, резките разместванията в икономиката настъпили от риск за здравето при COVID-19 пандемия от 2020-2021 г.

За инвеститорите целта не се постига по-лесно, за да са успешни е необходимо да се предвижда за дълго време напред икономическото развитие. Независимо, дали се инвестира в бизнес или друг актив, факторите за търсене и предлагане, и движението на цените, за да се излезе на печалба трябва да са близо до заложените в плановете. Тази група е един от основните двигатели за развитието на бизнеса в икономиката. Очевидно е, че добрите пазарни и ценови прогнози са от изключително важност за тях.

Групата на спекулантите е съществено по-различна от инвеститорската, защото тук се влагат голямо количество средства за възможно най-кратък период от време. Целта е максимална печалба от разлики в цените. Макар и група със сходен интерес както останалите, те са твърде различни участници на пазара. Рискът при тях е

изключително висок. Сделките за кратко време носят добри печалби или може да доведат до фалит. Често участието на спекулативни сделки може да доведе и до трусове в сектори на икономиката и финансите. Не е странно да се каже, че вероятно всеки един участник на пазара, би станал спекулант, ако има такава възможност. Дори може би и за кратко някога е бил. Но малък процент от участниците в тези сделки наистина успяват. За това тези с постоянен бизнес в този сегмент на икономиката са малък брой.

Очевидно е, че добрият финансов анализ на цените може да донесе значителни приходи за планиран период от време и е от изключителна важност за всеки. Ако резултатите от ценовите пазарни движения не покриват очакванията за всяка една от изброените икономически групи, може да се стигне до финансови загуби, фалит или замразяване на средства за неопределено време. Установено е, че пазарните цени подлежат на прогнозиране. За това са открити и съществуват различни методи (Plummer, 2010). Двата основни такива с които се работи най-често са технически (Plummer, 1991), (Scott, 2016) и фундаментален анализ (Wafi, 2015).

Фундаменталният се основава на анализиране на събитията, случващи се по света и касаещи финансовите и стокови пазари. Тълкуването им определя насоките за взимане на инвестиционни решения. Но негов недостатък е, че не предоставя точен алгоритъм, който да определи фундаменталното тълкуване. Възможно е според някой дадена новина да се счита за много важна, а пазарът да не го отрази по този начин. Също така да се очаква дълго една новина и когато тя стане факт да се окаже, че пазара вече я е изконсумирал. Успехът на фундаменталния анализ е изцяло зависим от качествата и интуицията на конкретен анализатор да тълкува точно събитията (Prechter, 2005). Техническият анализ е противоположност на фундаменталния, защото за да се направи анализ и изготви стратегия, не е необходимо да се следят новини и събития. Освен това, позволява методите за прогнозиране да бъдат описани точно, чрез статистически средства и математически алгоритми.

Съществуват мнения от утвърдени технически анализатори, че изобщо не бива да се влияем от новини (Prechter, 2016). Понеже чрез техническия анализ може даден пазарен сценарий да е разчетен като силно вероятен (Casti, 2002). Но според субективната преценка, заради новини да се действа емоционално и да се вземе грешно решение спрямо пазарната тенденция. Дори Тони Плъмър в книгата си за технически анализ „Прогнозиране на финансовите пазари“ (Plummer, 2010) има цяла глава с име

„Изключете телевизора“. От друга страна съществува и група от анализатори, които успешно обединяват двата метода фундаментален и технически. Те също твърдят, че на база фундаментално събитие ако се изгради бъдещ ценови сценарий, може да му се даде по-голяма тежест ако се потвърди и чрез технически анализ. Имало е много спорове, кой от двата подхода е по-добър.

В хода на развитието на финансовия пазар, голям брой проучвания показват, че пазарът е нелинеен и хаотичен. Особено по отношение на данните от финансовите времеви редове стойностите на акциите, чуждестранните борси и суровини на финансовия пазар, са чувствителни към въздействия и са склонни силно да се колебаят. Такива данни от времеви редове често имат силни нелинейни характеристики. По-добро прогнозиране на тенденцията на финансовия пазар е от голямо значение за намаляване на инвестиционния риск и вземане на финансови решения (Wu, 2020).

1.2.1 Невронни мрежи

Подходите за изследване на времеви редове могат да бъдат разделени на две категории: статистически методи и изчислителна интелигентност. Статистическите методи изследват зависимости между изходните и съответните фактори след изучаване на минали данни, докато другата група методи имитира човешкия начин на мислене и логическо заключение, за да придобие знания от миналия опит (като изкуствени невронни мрежи) и да предвиди бъдещи стойности.

В многобройната литература, посветена на времевите редове (Brockwell 1991; Brockwell 2002; Diggle 1990; Shumway 2006; Smith 2001), са описани важни стохастични методи за моделиране и прогнозиране на времеви редове. Подходът за изкуствени невронни мрежи (ANN) е предложен като алтернативна техника на прогнозирането на времеви редове и той придоби огромна популярност през последните няколко години. ANN са ориентирани към данни и са самоадаптивни по своя характер. При приложение на невронните мрежи не е необходимо да се посочва конкретна форма на модела или да се прави априорно предположение относно статистическото разпределение на данните; желаният модел се формира адаптивно въз основа на характеристиките, представени от данните. Този подход е доста полезен за много практически ситуации, при които няма налични теоретични индикации за подходящ процес на генериране на данни. ANN по своята същност са нелинейни, което

ги прави по-практични и точни при моделирането на сложни модели на данни, за разлика от различни традиционни линейни подходи, като ARIMA методи (Kihoro, 2004).

Изкуствените невронни мрежи (ANN) се използват дълбоко в различни научни и ежедневни задачи. Като инструмент за обработка на данни те са много добре познати от десетилетия (Azoff, 1994), (Silipo, 2013), (Oancea, 2013). Обикновено изкуствените невронни мрежи се представят като претеглен насочен граф и има много различни конфигурации на тази схема. В най-простия случай това е многослоен персептрон, но има много по-екзотични архитектури като обобщени мрежи (Tashev, 2002), например. В литературата са представени много различни обучителни алгоритми (Balabanov, 2018), но обратното разпространение на грешки за изчисляване на градиент, който е необходим за определяне на теглата в многослойния персептрон е най-популярният (Box, 2011). Обратното разпространение на грешката е точен числен метод, поради което има недостатък на вероятност за пропадане в локални оптимуми (Tomov, 2016). Чрез използването на стохастични или хибридни алгоритми за обучение се правят много опити за отстраняване на този недостатък (Atanasova, 2016), (Tudor, 2014), (Zankinski, 2017). Изборът на подходящи мрежови параметри е от решаващо значение при използване на ANN за прогнозиране. Също така, подходящото преобразуване или мащабирането на данните за обучение често е необходимо за постигане на най-добри резултати (Adhikari, 2013).

Времеви редове са атрактивни за изследвания с изкуствени невронни мрежи (Hill, 1996) (Gheyas, 2009), (Bernal, 2012), (Tomov, 2016). Традиционно методите за прогнозиране на времеви редове се основават на фиксирани линейни модели поради математическата им пригодност. Изследователите насочиха вниманието си към изкуствените невронни мрежи поради по-добрата им способност за апроксимация (George, 2019). Изкуствената невронна мрежа е универсален функционален апроксиматор като черна кутия от нелинеен тип, който е особено полезен при моделирането на нелинейни процеси, имащи априори неизвестни функционални отношения или системата от отношения е много сложна за описване. Провеждат се изследвания за моделиране на хаотични времеви редове (Hornik, 1989). Напоследък дълбокото обучение (Deep Learning - DL) привлече голямо внимание поради мощната си способност при разпознаване на модели и откриване на сложни структури в големи масиви от данни (Falat, 2015), (Li, 2020). Предлагат се хибридни невронни мрежи, като,

например, комбинация от стандартната RBF невронна мрежа, генетичен алгоритъм и пълзяща средна линия. Плъзгащата се средна линия се очаква да подобри изходите на мрежата, използвайки частта за грешка на оригиналната невронна мрежа (Falat, 2016).

Невронните мрежи предоставят обещаващ алтернативен подход за прогнозиране на времеви редове (Shamsuddin, 2008). Но не са разработени общи систематични процедури за изграждане на модели за прогнозиране с невронни мрежи (Shabri, 2001), (George, 2019). Ефективността на модела на невронни мрежи зависи от много фактори като естеството на данните във времеви редове, структурата на мрежите и процедурата за обучение (Zhang, 2012).

1.3. Времеви редове за подобряване на криптографията и кибер сигурността

Нормалното развитие на човечеството ни води до все по-голяма дигитализация. Все повече дейности и процеси са много по-продуктивни и ефективно управлявани чрез технологии. Тези процеси дори се ускориха и доказаха своята стойност, когато светът беше засегнат от глобалната пандемия COVID-19. Процесите, които биха отнели години, трябваше да се случат за месеци и обществото трябваше да търси нов начин на живот, който да е много по-свързан с технологиите. На пръв поглед изглежда, че светът е подготвен за подобно технологично предизвикателство. В известна степен това е така, но броят на киберпрестъпленията се е увеличил и посегателството върху лични данни, пари и загуба на информация, изнудване поради загуба на информация също ескалира до безпрецедентни нива. Всичко това е силен индикатор, че докато технологиите и изчислителната инфраструктура са посрещнали предизвикателството, ние не сме готови за силна киберсигурност (Jang-Jaccard, 2014), (Kostadinov,2019), (Dineva, 2019).

Изпълнението на изискванията за киберсигурност е предпоставка за безопасността и сигурността на ИТ инфраструктурите, цифровите ресурси и защитата на личните данни. В това отношение темите за криптографията и достатъчно стабилното генериране на случайни числа, които са в основата на всяка система за криптиране, представляват особен интерес.

За съвременните нужди на криптографията се използват два вида генератори на произволни числа - Генератор на случайни числа (RNG) и Псевдо генератор на случайни числа (PRNG) (DiCarlo, 2012).

Random Number Generator (RNG): прилага се при необходимост в даден момент от време RNG да генерира стойности, които трябва да са уникални и не трябва да се повтарят при следващи извиквания на RNG (Carr, 2003), (L'Ecuyer, 2007). Числата получени с този тип RNG се прилага към операции, които изискват уникални/неповтарящи се числови стойности генерирани във времето (Jin, 2004), (Camara, 2019). Пример за такава ситуация е генериране на криптографски ключ за кодиране/декодиране на данни, инициализиращи вектори, начални числови стойности за контролирани RNG и др (Ergün, 2015; Ryabko, 2016).

Pseudo Random Number Generator (PRNG): за основа на този генератор се използва първоначално число SEED. От тази стойност посредством алгоритъм произхождат всички генерирани в последствие случайни числа. Тези стойности по реда на тяхната поредност са ре-възпроизводими. Единствената неочаквана и тайна стойност, която трябва да е възможно най-непредсказуема е числото SEED, което е „корен" в основата на тази числова редица и основа за генериране на цялата числова редица. От тази технология е взаимствано използването най-често за удостоверяването с One Time Password (OTP), генерирането на криптографски ключове произлизащи от Master Root Key, което е се прилага при съставянето на портфейли в Block Chain - distributed ledger technology, удостоверяване чрез HMAC и др.

Традиционните мерки за RNG са предимно обобщена статистика, отнасяща се до отклонения от математическата случайност (Trappe, 2006).

Хардуерният генератор на случайни числа (HRNG) (Dichtl, 2003) или още истински генератор на случайни числа (TRNG) е устройство, което генерира случайни числа от физически процес, а не посредством алгоритъм. Този тип генератори са коренно различни от разглежданите до тук. Защото такива устройства често се основават на микроскопични явления, които генерират ниско ниво, статистически случайни сигнали "шум", като топлинен шум, фотоелектричен ефект, включващ разделител на лъча и други квантови явления. Тези стохастични процеси на теория се смятат за напълно непредсказуеми, за разлика от парадигмата за генериране на псевдослучайни числа, често прилагана в компютърни програми. Като цяло са известни два основни източника на практическа квантово-механична физическа случайност: квантовата механика на атомно или субатомно ниво и термичен шум (някои от които са с квантово-механичен произход). Квантовата механика прогнозира, че някои физични

явления, като ядреното разпадане на атомите, са фундаментално случайни и по принцип не могат да бъдат предсказвани.

Тъй като резултатът от квантово-механичните събития не може да бъде предвиден дори по принцип, те се водят за „златният стандарт“ за генериране на случайни числа. Всъщност едни от най-добрите генератори за случайни числа за сървърни системи се смятат квантови генератори от фотонен тип. Защото са достатъчно компактни и могат да се поберат върху платка за вграждане в компютър и са с много висока производителност. Според резултати от някои изследвания дори е много вероятно един такъв хардуерен модул от този тип, да има капацитет да захрани с качествени случайни числа повече от един сървър за публични услуги.

Независимо кой от генераторите на произволни числа е необходим (неконтролиран или контролиран), общият успех на системата зависи от качеството на произведените генерирани произволни числа (Lavasani, 2009). Бързо нарастващото търсене на честотна лента, обеми на съхранение на данните и изчисления, съчетано с нарастващия спектър от киберзаплахи гарантират, че нуждата ни от надеждни и непредсказуеми случайни числа само ще расте в бъдещ (Hart, 2017).

Темата за RNG и PRNG заслужава особено внимание, особено защото криптографската защита в информационните системи разчита на нея, а и в други технологични области (Fan, 2018), (Li, 2020) са необходими добри RNG решения. За това е изключително важно да сме сигурни в качеството на система или механизъм за генериране на произволни числа. Следователно, това доказателство трябва да е резултат потвърден от повече от една система за представяне и анализ на случайните числа. Комбинирането на различни подходи ще доведе до по-точни и точни резултати и заключения.

1.4 Изводи

В резултат на направения аналитичен обзор могат да бъдат изведени следните заключения:

Въпреки многобройните постижения, задачите за анализ на данни при изследване и прогнозиране на динамични явления в областта на разнообразни информационни системи все още са предизвикателни поради високата сложност на тези системи. Изследванията на времеви редове в различни области и приложения се

нуждаят от разработка на специфични методи и средства за постигане на конкретните цели.

1.5. Цел и задачи на дисертацията

Целта на настоящата дисертация е да се разработят нови методи и средства за анализ на данни в информационни системи с използване на времеви редове.

За тази цел се дефинират следните задачи:

1 да се разработи метод за анализ и предсказване на ценови движения във финансовата област с използване на времеви редове;

2 да се предложи алгоритъм за обучение на изкуствени невронни мрежи при прогнозиране на финансови времеви редове;

3 да се предложат решения за повишаване на криптографската защита в информационните системи чрез прилагане на методи за анализ на времеви редове;

4 да се проведат експериментални изследвания за верификация на предложените методи за повишаване на криптографска защита при решаване на задачите за осигуряване на киберсигурността.

5 Да се разработят програмни методи за преодоляване на проблеми при работа с големи обеми от данни във времеви редове.

Глава 2. Методи за изследване и прогнозиране на финансовите времеви редове

2.1 Подобряване точността на Моментум в комбиниране с един метод за прогнозиране на пазарни ценови движения

След представените подходи за анализ и предсказване на ценови движения в първа глава тук вниманието се насочва към техническия анализ. В тази глава ще бъде разгледано изследване върху широко разпространения индикатор *Моментум*, който принадлежи на групата на осцилаторите. Изчисляването му се базира на математически апарат за обработка на времеви редове. Анализът има за цел да даде по-ясна представа за предимствата и недостатъците на този широко разпространен осцилатор и да подобри неговата ефективност.

2.1. 1 Осцилатор Моментум (Momentum Oscillator)

Моментум е основен индикатор, който показва дали ценовата тенденция се ускорява, забавя или се движи със същата скорост. Той обикновено достига максималната си стойност преди върха на цените и минимума преди дъното на спада. В скалата на този индикатор се намира неутрална линия 100. Когато се получи стойност над тази линия се приема, че тенденцията е възходяща, а когато се намира под тази линия се приема, че властва низходящата тенденция. Формула на *Моментум* е следната:

$$M = P_t - P_{t-n}$$

където: M – момент; P_t - цена на затваряне към настоящия момент от време; P_{t-n} - цената на затваряне преди t-n период от време;

Броя на времевите интервали n се избира от потребителя. Например при използването на дневна времева рамка, 10-дневния Моментум на цените е равен на разликата между сегашната цена и цената преди 10 дни. Съответно при промяна на времевата рамка към часова или минутна, стойностите се взимат според времевото отмерване. Моментум е положителен, ако сегашната цена е по-висока от предишната и отрицателен, ако сегашната цена е по-ниска от предишната и равен на 0, ако двете цени

са равни. Наклонът на линията, която свързва точките на Моментум, изчислен за всеки ден, показва дали нараства или намалява. В платформите за анализ и търговия, където е интегриран индикаторът, тези формули се изчисляват автоматично от системата (примерни такива: Meta Trader 4, Meta Trader 5, Bloomberg Terminal, Trading View, Trader Workstation и др.). Там на потребителя (анализатор) се представя готовата графика (Person, 2007), както е показано на фиг. 2.1, която представя ценовото движение на анализираната валутна двойка на дневна база. Осцилаторът Моментум е изчертан с криволичеща линия в отделена секция в долната част на изображението (отбелязано с *1A*), която се движи около справочна хоризонтална права линия (отбелязана с *1B*). Областта на хоризонталната ос *1B* е със стойност 100 и е ключова за откриване на ценовите тенденции. В случай, че последната цена на затваряне е по-голяма от n-периода назад, стойностите на Моментум ще бъдат разположени над нея (*1B*), а в обратния случай под оста *1B*. Пробивът на средната линия се явява, като

сигнал за покупки или продажби.

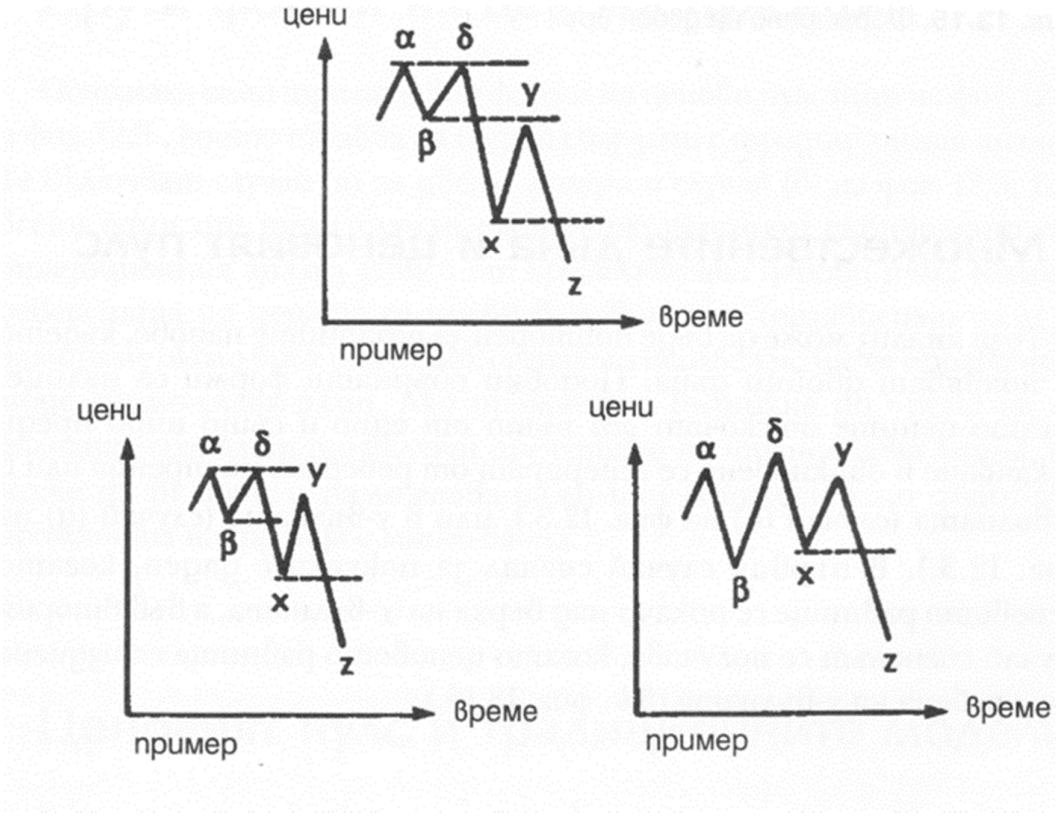
Фиг. 2.1. Ценовото движение на анализираната с Моментум валутна двойка EUR/USD между септ. 2006 г. до август 2008 г., дневна времева рамка (исторически данни от Forex пазарът)

Дивергенцията се счита за най-силна при този инструмент за прогнозиране. Тя може да бъде възходяща и низходяща.

Когато цената достига по-високи върхове, а линията на индикатора отказва да отчете по-висока стойност. Това означава, че според Моментум трендът се намира във фаза на възходяща дивергенция. Аналогичен е случаят при низходящата дивергенция, но тогава тя се изразява в по-ниски дъна и стойности на Моментум все по-близко разположени от долната страна на средната линия. При наличието на този сигнал може да се смята, че достигането на цената на нов пазарен връх е по-малко вероятно. В този случай една по-дълбока корекция на тренда или продължително движение настрани може да е достатъчно и пазарът да набере сила, за да подмине предходните стойности в които е отчетена дивергенцията. Пример за възходяща Дивергенция може да се наблюдава на Фиг.2.2. Цената на пазарния тренд в горната част на Фиг.2.2 чертае по-високи върхове, но линията на осцилатора в долната част на Фиг.2.2 се приближава все по-близо до средната ос, като отказва да направи по-висок връх спрямо нея. Дивергенцията е подчертана на графиката на осцилатора с линия *1C*. Там е видно, че стойността на EUR спрямо USD в посочения период е имала най-висок пик през 2008 г. Но също така личи, че след отчетената дивергенция, цената отказва да направи нов връх за определен период и може да се забележи, че дори навлиза в забавяне и по-дълбока корекция. Като сценарият за тази тенденция се потвърждава и от още едно явление описано в техническият анализ, като множествен връх – отбелязан на Фиг.2.2 с *1D*.

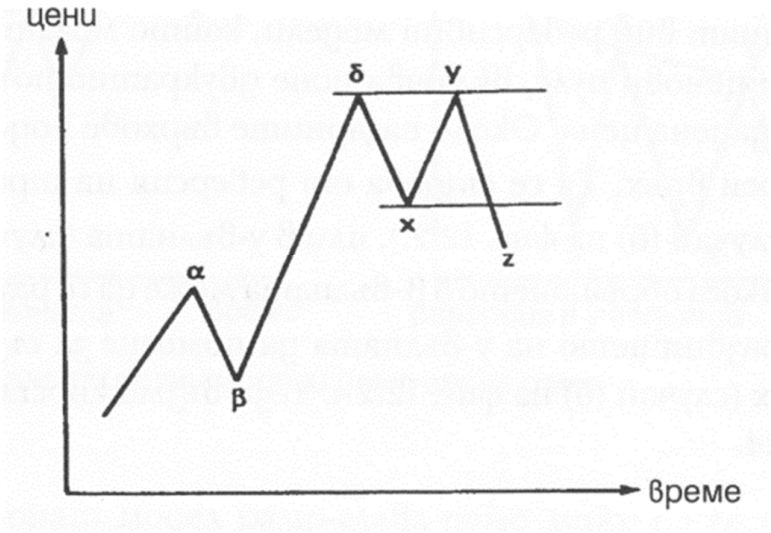
Фиг. 2.2. Дивергенция на Моментум при пазарен тренд при валутна двойка EUR/USD между септ. 2006 г. до август 2008 г., дневна времева рамка (исторически данни от Forex пазарът)

Множествен връх е формация от вида реверсивни модели, която може да се появи в единичен ценови пулс, включва поне двукратното отскачане на пазара от дадена цена. Около пазарните върхове, като в подобна форма се нарича още двоен връх (Bulkowski, 2000). Тази формация се създава от реверсия на тренда или в β-вълната (фиг.2.2.1а.), или в γ-вълната (фиг.2.2.1а.). Понякога обръщането в (β-вълната може да се развива толкова бавно, че развитието на γ-вълната да помогне за създаването на троен връх (фиг.2.2.1а.). Тези възможности са представени на Фиг.2.2.1а.



Фиг.2.2.1а. Формиране на множествен връх (Plummer T.,1991)

От друга страна се смята, че реверсията през у-вълната е много по-леко за идентифициране и дава лесно уловим сигнал за продажба, когато цените паднат под нивото на х-вълната (фиг.2.2.1б):



Фиг.2.2.1б. Формиране на двоен връх (Plummer T.,1991)

Този анализ може да бъде повторен за пазарните спадове, където се появяват двойни дъна. Подобни обърнати форми са налице, когато цените отскочат два пъти от едно и също ниво преди покачване и обикновено се генерират от реверсия на тренда или в β-вълната (случай (б) на фиг.2.2.1в), или в γ-вълната (фиг.2.2.1в). В първия случай сигнал за покупка е даден, когато ценовото равнище се покачи над върха на α-вълната, а във втория случай сигналът се получава, когато ценовото равнище се издигне над гребена на γ-вълната (вж. фиг.2.2.1в).

Фиг.2.2.1в. Формиране на двойно дъно (Plummer T.,1991)

Формирането на описаният модел на повтаряемост двоен връх е отбелязан на Фиг.2.2 с елипса и е дефиниран като *1D*. В посоченият случай се явява, като потвърждаващ сигнал за понижаването на цената, което е посочено и от дивергенцията

на Моментум и в последствие се наблюдава на същата графика като по-дълбока корекция. В техническия анализ се смята, че за да се дава по-голяма тежест на вероятен пазарен сценарий е необходимо очаквана тенденция на да се потвърди с повече от един инструмент за анализ. Но дали наистина участник в пазара, който се доверява само на един инструмент за прогнозиране, би сгрешил?

2.1.2 Слабости при анализ на пазарния тренд чрез Моментум

Наистина в повечето случаи този осцилатор работи прекрасно и е добър предвестник на бъдещи промени в текущата пазарна тенденция. Това го е утвърдило като инструмент, част от арсенала на борсови търговци и експертни системи за борсова търговия на инвестиционни фондове. Но ако се анализира продължително време пазара чрез Моментум, може да се забележат известни, макар и не често срещани изключения.

Фиг. 2.3. Валутна двойка USD/CAD на дневна база между 1997 и 1999 г. (исторически данни от Forex пазарът)

На фиг.2.3 се вижда, че цената е начертала един последен най-висок връх към дата 11.04.2013 г. След което трендът навлиза в дълбока и продължителна корекция. Но, анализирайки поведението чрез Моментум, може да се допусне, че ще последва още покачване за американския долар, защото осцилаторът не навлиза в дивергенция, а продължава да нараства. На графиката Фиг.2.3 в долната част на екрана, явлението е маркирано с линия 1С. Там линията на движението на Моментум е начертава връх по-

висок даже от предходния, но цената след това е направила значителна корекция от около 60% без да е на лице дивергенция при осцилатора. Но дори да се сметне, че това е временно положение и има изгледи тренда да запази възходящата си тенденция в много дългосрочен план, корекция от този мащаб може да доведе до фалит или да блокира значителни финансови средства за дълъг период от време. Уповавайки се на правилото, че за да се вземе решение за покупка на тези нива, е необходимо да има поне още един сигнал за потвърждение от различно средство на техническия анализ. Следователно не би трябвало да се предприеме действие за покупки, но ако единият инструмент на който се разчита е Моментум, то вече е на лице грешен сигнал за вземане на решение. Тук наистина не е необходимо да се чертаят повече вероятни сценарий за погрешни инвеститорски решения. Въпросът, който вълнува изследването е, дали може да се подобри прецизността на Моментум?

2.1.3 Метод за повишаване точността на Моментум

Наистина хубаво би било, ако се знае повече за състоянието на пазара в даден момент. Като например, дали по-голямата част от участниците, са купили или продали и няма много останали от тях, за да придвижат още цената в определена посока. Такъв е случаят, когато на пазарът властва краен сантимент. Което всъщност е причината за ускорението на Моментум заедно с цената в дадена посока, без да се засече забавяне и разликите необходими за наличието на дивергенция отчитана от осцилатора. Но отчитането на това събитие след, като то е приключило не предпазва от вземането на погрешни решения, а само ги обяснява. Освен това, според психологията на участниците в търговията, продължителното движение на цената в дадена посока също често пъти подвеждащо. Такива крайни тенденции, като на фиг.2.3, може да създават чувство на сигурност и оптимизъм в участниците за още по-продължителното покачване на цената. Много пъти е наблюдавано, че до последния момент преди настъпи обрат, сантимента на участниците е бил крайно оптимистичен. Но какво би се случило, ако и точно тогава средства за анализ ни подвеждат?

Като възможно решение на проблема, може да се разгледа пазарен индикатор изграден на базата на пълзящата средна линия (MA):

Какво представлява Moving Average (MA)?

Moving average са един от най-старите и прости методи. Отразяват осреднена цена за определен период от време. Делят се на два типа: Simple moving average (SMA) и Exponential moving average (EMA). (Dahlquist, 2006)

Simple moving average (SMA)

При пресмятане на периода на SMA, се използва времеви ред, при който се сумират данните на последните периоди (*t*), където например *t*=10 за 10 дена, според времевата рамка (може да бъде различна стойност, по избор). След това се дели на броя *t* периодите. Такова пресмятане се прави за всеки един бар за период от графиката. Формулата за SMA е, както следва:

$$SMA_t = \sum_{n=1}^{t} price_n / t$$

Като пример : 5 дневна SMA ще се пресмята чрез събиране на цените при затваряне на борсата за последните 5 дни и делене на цялата сума на 5.

10 + 11 + 12 + 13 + 14 = 60;

60 / 5 = 12

MA се „плъзгат“, защото новият период се добавя, а най-старият период се изоставя. При следваща цена на затваряне средната е 15, след това най-новият период 15 ще бъде добавен, а най-старият ден, който е 10, ще бъде изоставен. Новата 5 дневна MA ще се пресмята както следва:

11 + 12 + 13 + 14 + 15 = 65;

65 / 5 = 13

През последните 2 дни MA се е изместила от 12 на 13. След добавяне на новите дни, най-старите ще бъдат извадени и MA ще продължи да се придвижва.

В примера по-долу е използвана цената при затваряне на Eastman Kodak (EK), 10-я ден е първият възможен за пресмятане на 10-дневната MA. При 10-дневната MA, за 11-я ден пресмятането ще се извърши по следния метод – добавяне на цената от 2 до 11 ден, разделено на 10. Същият процес се повтаря и при пресмятането на 12 ден.

- пресмятат се цените от 3 до 12 ден и се делят на 10.

Day	Daily Close	10- day SMA
1	67.50	
2	66.50	
3	66.44	
4	66.44	
5	66.25	
6	65.88	
7	66.63	
8	66.56	
9	65.63	
10	66.06	66.39
11	63.94	66.03
12	64.13	65.80
13	64.50	65.60
14	62.81	65.24
15	61.88	64.80
16	62.50	64.46
17	61.44	63.95
18	60.13	63.30
19	61.31	62.87
20	61.38	62.40

Фиг.2.4 Стойности и изчертаване на SMA (исторически данни от пазарът на акции)

Графиката на (вж. Фиг.2.4) съдържа част от структурираната информация в горната таблица. SMA започва от 10-я ден и продължава във времето. Тази проста илюстрация скрива най-важния факт, поради който МА са изоставащи индикатори и те винаги са след цената. МА са изоставащ индикатор, който определя тенденцията на движение. Когато цената съвпада с направлението на движението.

Exponential moving average (EMA):

С цел да се намали изоставащия ефект на Simple moving average, ползващите технически анализ често предпочитат Exponential moving average (EMA). Те намаляват изоставането чрез добавяне на нови стойности върху най-новите цени, зависещи от дължината на МА. Най-кратката ЕМА ще е с по-голяма стойност, отколкото ще бъде приложена за повечето МА. За пример: 10-дневната ЕМА има стойност при добавянето към цената – 18.18%, а 20-дневната ЕМА, в повечето случаи, има стойност добавена към цената – 9.52%. При пресмятане на ЕМА, най-важно е да се запомни, че тя добавя допълнителна стойност към последната цена. Следователно, ще реагира по-бързо на промяната на по-новата цена, отколкото SMA.

Формула за пресмятане на ЕМА:

$$X = (K \times (C - P)) + P$$

X – настояща ЕМА

C – настояща цена

P – EMA от предния период *

K – изглаждащ коефициент

* - (за пресмятането на първия период се използва стойност от SMA)

Изглаждащият коефициент прилага подходящ коефициент към по-новите цени, които са свързани с предходните цени на EMA. Формула за изглаждащия коефициент:

K = 2 / (1+ N)

N – брой на предходните EMA цени

Формулата на EMA работи чрез определяне на разликата между цената на сегашния период и цената на предходния период на EMA. Има два възможни изходни резултата : разлика към положителна или отрицателна стойност.

1. Ако сегашната цена е по-висока от предходния период на EMA, разликата ще бъде положителна (C - P). Положителната разлика е съответно умножение на положителната разлика с изглаждащия коефициент ((C - P) K) и отговорът е добавен към резултата от предходния период на EMA; резултатът е по-висока EMA

(K (C – P)) + P

2. Ако сегашната цена е по-ниска от предходния период на показателната EMA, разликата ще бъде отрицателна (C - P). Отрицателната разликата е съответно умножение на отрицателната разлика с изглаждащия коефициент ((C - P) K) и отговорът е добавен към резултата от предходния период на EMA; резултатът е по–ниска EMA (K (C – P)) + P.

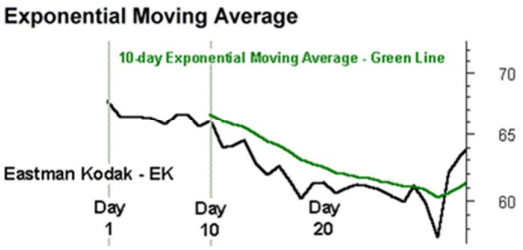
По-долу е показана таблицата с резултати от EMA на Eastman Kodak. За първия период на EMA се използва SMA като предходна стойност на EMA. От 11-ти период напред се използва предходната цена на EMA. Пресмятанията на 11-ти период са както следва:

1. (C - P) = (73.81 – 74.28) = -0.47;

2. (C - P) * K = -0.47 *0.1818 = -0.08

3. ((C - P) *K) + P = -0.08 + 74.28 = 74.20

Day	Current Price (C)	Previous periods EMA (P)	10-day EMA (X)	EMA Periods (N)	Smoothing constant (K)
1	76.69				
2	76.13				
3	75.50				
4	74.94				
5	74.84				
6	75.19				
7	73.81				
8	73.38				
9	73.06				
10	72.75	74.63	74.28	10	0.18182
11	73.81	74.28	74.20	10	0.18182
12	75.63	74.20	74.46	10	0.18182
13	75.25	74.46	74.60	10	0.18182
14	76.94	74.60	75.03	10	0.18182
15	76.38	75.03	75.27	10	0.18182
16	76.31	75.27	75.46	10	0.18182
17	75.44	75.46	75.46	10	0.18182
18	74.75	75.46	75.33	10	0.18182
19	74.69	75.33	75.21	10	0.18182
20	72.44	75.21	74.71	10	0.18182



Фиг.2.5 Стойности и изчертаване на EMA (исторически данни от пазарът на акции)

Разликата между EMA и SMA е минимална, но въпреки всичко съществува. В края на последните 10 периода от времевия ред, EMA е по-близо до пазарната цена, отколкото SMA. SMA е по-близо до пазарната цена само в период от времевия ред 18, но това не продължава дълго. Средната абсолютна разлика между EMA и пазарната цена е - 1, а SMA е със средна абсолютна разлика - 1,33. Това ще означава, че EMA е средно с 1 точка под или над пазарната цена и че SMA е с 1,33 под или над пазарната цена. Тази разлика може да се забележи в графичен вид на фиг.2.4 изчертана с червено EMA и SMA със зелено от фиг.2.5. Пълзящите средни линии са приложени върху едни и същи данни на Eastman Kodak от пазарът на акции.

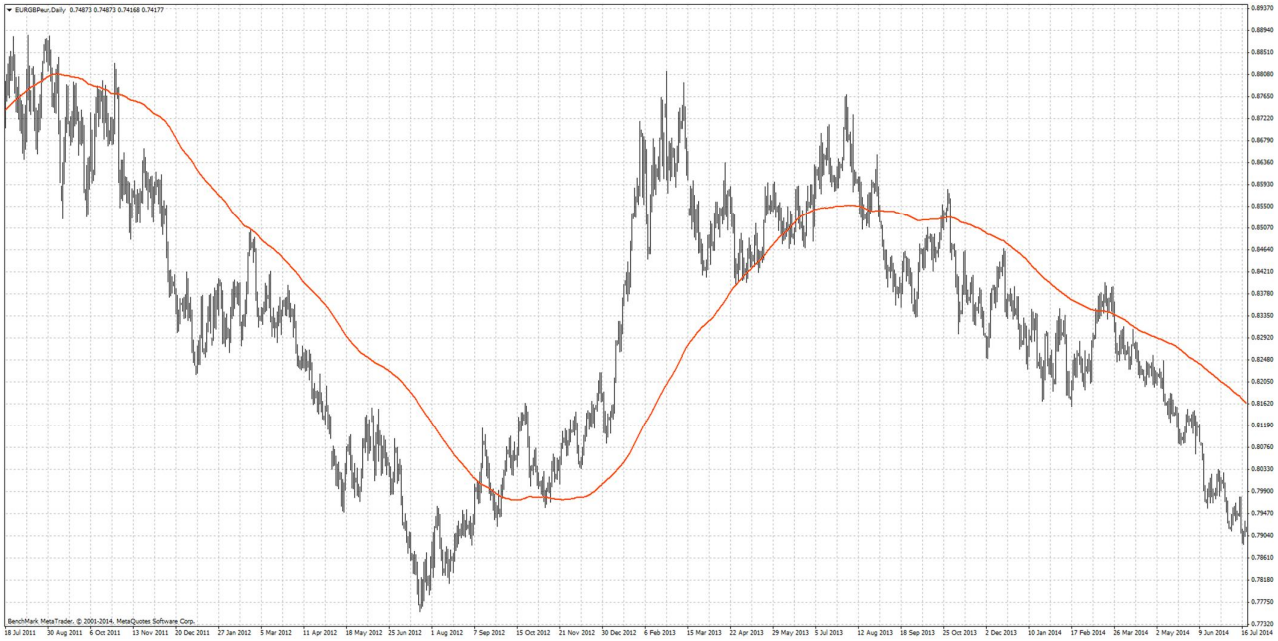
Система за търговия чрез MA:

Използване на този индикатор в търговска система, се базира на повече от една пълзяща средна линия, като по предпочитания, може да бъде от различен вид. Използването на две MA би давало сигнал за купуване, когато по-късата (по-бързата) MA напредне над по-дългата (по-бавната) MA. Сигнал за продаване ще бъде получен, когато по-късата MA пресече надолу по-дългата. Скоростта на системата и броят на генерираните сигнали ще зависи от дължината на MA. Системите с по-къси MA, ще генерират повече сигнали и ще бъдат по-пъргави. Обаче биха генерирали и повече грешни сигнали от системите с по-дълги MA. Този подход е приложим, както при системи за търгуване от човек, така и в автоматизирана система за търговия.

Друга известна система е само с една MA, която е с по-голяма стойност на по-висока часова рамка. Като пазарният анализатор следи, дали цената все още се движи

над МА при възходящ тренд или под нея при низходящ. Ако цената трайно се задържа от едната страна линията, се счита за продължаване на текущата тенденция. При доближаване на цената и „отскачане“ от МА, като цената продължи да се придвижва в същата посока, също се приема за сигнал за потвърждение на текущата тенденция. В случай на пробив под или над МА, анализаторът приема това за сигнал за възможно преобръщане на тенденцията или за навлизане във фаза на по-дълбока и продължителна корекция (фиг.2.6). В този подход, без комбинация с допълнителен инструмент (индикатор, осцилатор или друго подпомагащо изчисление), не е възможно да се автоматизира в система за вземане на решение. В този вид подходът се използва предимно за човешки анализ.

Определено МА могат да бъдат ефективни инструменти за дефиниране и потвърждаване на направление, определяне на поддържащи и съпротивителни нива на цените, като често подпомагат разработка на много търговски системи. Предимствата, при използване на МА трябва да се преценят спрямо недостатъците. МА следват направленията - индикатори, които винаги ще са една стъпка назад. И тук, като при останалите инструменти за технически анализ, важи правилото, че МА не бива да се използват самостоятелно, а в съчетание с други инструменти, с които да се съвместяват.



Фиг.2.6. 21 MA при валутен тренд на EUR / GBP между 2011 и 2015 г. на дневна времева рамка (исторически данни от Forex пазарът)

В търсене на решение на изложения в изследването проблем с Моментум, се включва и MA. Като много е важно да се избере правилната стойност на времевия *(t)* ред за пълзящата средна линия. Не трябва да е твърде „бърза“ или при по-големи стойности да изостава твърде много от цената. От продължителен анализ за нуждите на това изследване се оказа, че на дневна времева рамка, 21 дневната стойност за MA е достатъчно балансирана и дава достатъчно добри резултати (фиг.2.6). И за да се разбере по-добре, какво представлява предлаганото решение, нека се върнем на идеята за махалото и си припомним, че финансовите пазари са циклични. Циклите могат да бъдат малки, големи, могат да се обяснят чрез физични явления или да не могат, а да са дефинирани от продължително наблюдение на статистически повторения. Ако се приложи този подход и се направи аналогия на поведението на цената към пълзящата средна линия, може да се каже, че цената не би могла да се отдалечи твърде много от MA без в един момент отново да се върне отново до нея.

Прилагайки в изследването този начин на мислене, се стигна до следната абстрактна интеграция, която избираме да наричаме с името MA Volatility Indicator. Да си представим, че за наблюдавания период и движение MA е водеща, а пазарният тренд е вторично явление, което зависи от посоката на MA. Следователно може да се дефинира, че пазарният тренд е обект, който гравитира около MA, заради някаква невидима притегателна сила. При което, ако твърде много се отдалечи от MA, ще последва силно обратно завръщащо движение към нея. Както махалото се връща, след като отслабне приложената върху него сила. За отклонението на пазарната цена от MA, може да се намерят стойности на отдалечаване, които да са критични. При достигане на критичните стойности, може да се очаква силно завръщане към водещата линия (MA). От продължителни наблюдения може да се потвърди, че и тук, както при махалото, колкото по-силна е била енергията за отдалечаване на цената от MA, толкова по-силно ще е обратно движение което се очаква. Което може да се превърне и в по-дълбока и продължителна корекция на текущата тенденция. За изобразяване на това явление върху пазарен тренд, виж фиг.2.7. Там е представена реална графика на пазарно движение, където е отбелязано в пазарни пипсове отмереното разстояние на пиковете на отдалечаване на цената от MA.

Фиг.2.7 Отдалечаване на цената от MA измерено в пазарни пипсове (исторически данни от Forex пазарът GBP/AUD в период януари – август 2012 г.)

От стойностите на фиг.2.7 може да се опишат следните резултати:

- отклонение на тренда между 1 и 400 пипса в посоката на движение на MA не е твърде екстремно и е възможно текущата тенденция да продължи;
- при стойности на отдалечаване на тренда по-големи от 400 пипса от MA, може да се считат за екстремуми и тяхното достигане трябва да се приеме за сигнал за по-дълбока и продължителна корекция или обръщане на тенденцията.

След достигане на екстремните стойности на отклонение от MA, пазарната цена продължава за кратко време движението си в същата посока. Тези места сигнализират за висок сантимент сред участниците на пазара и са свръх покупка или свръх продажба. Такова явление води до саморегулиране на ценовата тенденция при свободния пазар. За това след него може да се наблюдава корекция с продължаване на текущия тренд или преобръщане на тенденцията.

Този подход може да се ползва, както за откриване обръщане на тенденцията или търсенето на по-дълбоки и продължителни корекции, така и за потвърждение на текущата тенденция. На фиг.2.7.1 се вижда запазване на текущата тенденция, като цената се отдалечава от MA, без да достига екстремни стойности, както при фиг.2.7. При това отдалечаване, цената се връща до MA и продължава да следва същата

тенденция. Скоростта на тенденцията се запазва и това може да се използва, като сигнал за запазване посоката на движение на пазара.

Фиг.2.7.1 Отдалечаване на цената от MA (исторически данни от Forex пазарът USD/CHF в периода август 2011 г. до юни 2012 г.)

Това явление може да се обяснени от една сравнително новата наука наречена социономика, на която автор и основоположник е Робърт Пректър (Prechter, 2016). Той е американски борсов анализатор, известен е със своите точни пазарни прогнози. Към настоящия момент е автор и съавтор на 14 книги за финансовите пазари. Някой сред най-известните прогнози, които е правил са кризата настъпила 2007 г., както и обратът в цената на златото през 2012 г. При двата случая се аргументира много добре, като настъпващата криза 2007 г., освен в множество интервюта я предвижда и в една от книгите си, като много точно описва причините за нейното настъпване. За дълбокия и продължителен спад в цената на златото също се аргументира, като описва пазарът като свръх прекупен. Чрез анализите сочи твърде високия сантимент в участниците и определя пазара, като свръх прекупен и счита, че сега е момента за продажби и намаляване на експозициите от този пазарен инструмент. Прогнозата се оказва отново много точна, цената на златото пада с около 70% и влиза в продължителна 6 годишна корекция до 2018-та г., където отново се завръща във възходяща си тенденция.

Чрез науката социономика се предлага сериозна аргументация за случващите се възходящи и низходящи движения на цените. Анализира се движението на цените и се

дава обяснение за настроенията и нагласите на пазарните участници. В двата тома на книгата Socionomics: The Science of History and Social Prediction (Prechter, 2003) се предлага едно последователно и интердисциплинарно обяснение на сложните процеси, които протичат в обществения живот, използвайки изключително разбираеми примери. Социономическата теория (Prechter, 2016) е първата, която дава логично обяснение на факта, че пазарните индекси изпреварват икономическата действителност. Факт, открит още преди век, но отговорът на въпроса защо се получава така е предложен от Р. Пректър едва в наши дни.

Отново е време да се насочи вниманието към Моментум и да се види причината за откритият проблем и негово възможно решение, чрез комбиниране с MA Volatility Indicator. Оказва се, че случаите при които Моментум прави своето изключение с дивергенцията, са точно местата на които пазарът се намира в екстремен сантимент според приложеното изследване озаглавено MA Volatility Indicator. Следователно, ако Моментум се комбинира с предлагания в изследването метод, има възможност да се повиши неговата прецизност. На фиг. 2.8 е представен реален пример, при който са комбинирани сигналите от осцилатора и са изведени стойностите на отдалечаване цената от MA.

Фиг.2.8 Комбиниране на Моментум с предлагания метод - MA Volatility Indicator (исторически данни от Forex пазарът USD/CAD на дневна база между 1997 и 1999 г.)

На графика фиг.2.8 е отмерено, че в даден момент от време и спрямо MA, цената достигнала отдалечаване на 554 пипса от 21 дневната MA. Което е стойност над

откритите критични стойности при този пазарен инструмент от 400 пипса. Според предлагания в тази дисертация метод (MA Volatility Indicator) за измерване на сантимента, чрез отдалечаване на цената от MA, това явление може да се дефинира, като сигнал за наличието на екстремен сантимент в участниците и вероятност от по-дълбока ценова корекция. В последствие се вижда, че тенденцията прави силен спад и коригира до стойности от 1.4440 CAD за USD. Това е потвърждение на факта, че последният достигнат връх е бил ниво на свръх покупка. Въпреки, че Моментум не го отчита с дивергенция, реално нивото на върха от 1.5854 CAD за USD е било твърде прекупено за този момент от време.

След представените резултати, може да се каже, че заради изключението което понякога се генерира, Моментум наистина не отбелязва винаги с дивергенция опасни пазарни зони. Също така след различни случай на наблюдение на посоченото явление става ясно, че то се случва само в моменти на краен пазарен сантимент. Което се обяснява от факта, че в този момент пазарът е в пик с висок екстремум, но осцилаторът няма ограничение в движението си и не предсказва добре в такава ситуация. Според начина му на прилагане, би трябвало да се очакват нови по-високи нива на цената, при които вече ще е наличен сигнал за дивергенция. Но в комбинация с предлаганото решение, може да се направи надежден анализ и да се достигне до по-точна прогноза на критичните периоди.

В заключение може да се каже, че Моментум е наистина един добър, доказал се във времето и използван осцилатор със задоволителен процент успеваемост, заради което е намерил място сред много пазарни анализатори и трейдъри, а също и сред множество автоматизирани системи за пазарна търговия. Всеизвестно е, че не съществува универсален метод за анализ, който на 100% да постига точност при прогнозиране на бъдещи пазарни тенденции, като винаги съществува риск от загуби. Но от резултатите в това изследване, може да се направи заключение, че макар и Моментум да има някои недостатъци, ако се комбинира с предлагания в изследването метод може да се постигнат по-добри резултати. Комбинацията от тези два инструмента за пазарен анализ би могло в бъдеще да намерят място и в някои специализирани платформи за търговия, което да спомогне за намаляване на риск от загуби и увеличаване процента на успех при търговията. Също така е възможно предлаганото решение да се интегрира към системи за автоматизирана търговия и вземане на решение.

2.2 Прогнозиране на финансови времеви редове чрез невронни мрежи

Прогнозирането на финансови времеви редове е привлекателна област за изследвания (Maciel, 2009), (Balabanov, 2011). Много популярен подход в тази област е използването на изкуствени невронни мрежи. Изкуствените невронни мрежи са разработени като математическо обяснение на биологичните невронни мрежи, но тяхното приложение е все още далеч от живите организми. Изкуствените невронни мрежи имат две общи работни фази. Първо се обучават и след това се използват за решаване на конкретната задача. Многобройни проучвания показват, че добре обучената мрежа може да реши своите задачи много ефективно. Обучението е ключов елемент, защото отнема време дори на много мощни компютри (Balabanov, 2017). Някои изкуствени невронни мрежи са ориентирани към дълбокото обучение като алгоритъм за обучение. В това проучване вместо удължаване на номера на скрити слоеве се увеличава размерът на входящия слой на трислойния многослоен персептрон.

Многослойният персептрон е най-често използваният вид на изкуствени невронни мрежи, който може да се представи като ориентиран претеглен граф. Възлите на графа се наричат неврони. Връзките между невроните имат тегла и тези тегла са в основата на информацията, представена в мрежата.

Изкуствените невронни мрежи работят в два общи режима - обучение и работа. Режимът на обучение се изпълнява като задача за оптимизация, при която тежестите в мрежата трябва да бъдат модифицирани по такъв начин, че мрежата да научи най-добре моделите на обучение. Има много алгоритми за обучение, разработени през последните четири десетилетия, но най-популярният е обратното разпространение на грешката. Обратното разпространение на грешката е точен числен метод и това е предпочитаният метод на обучение в това проучване. Идеята е свеждане до минимум на общата грешка на невронната мрежа, постигната по време на обработката на всички примери за обучение. Градиентът на общата грешка се използва за актуализиране на теглата като посока на актуализацията и величина на актуализацията. Начинът, по който се организират връзките между невроните, е общ за топологията на изкуствената невронна мрежа. Има много различни топологии, широко изследвани в литературата, като генерализирани мрежи (Tashev, 2003) или дълбоки обучителни невронни мрежи.

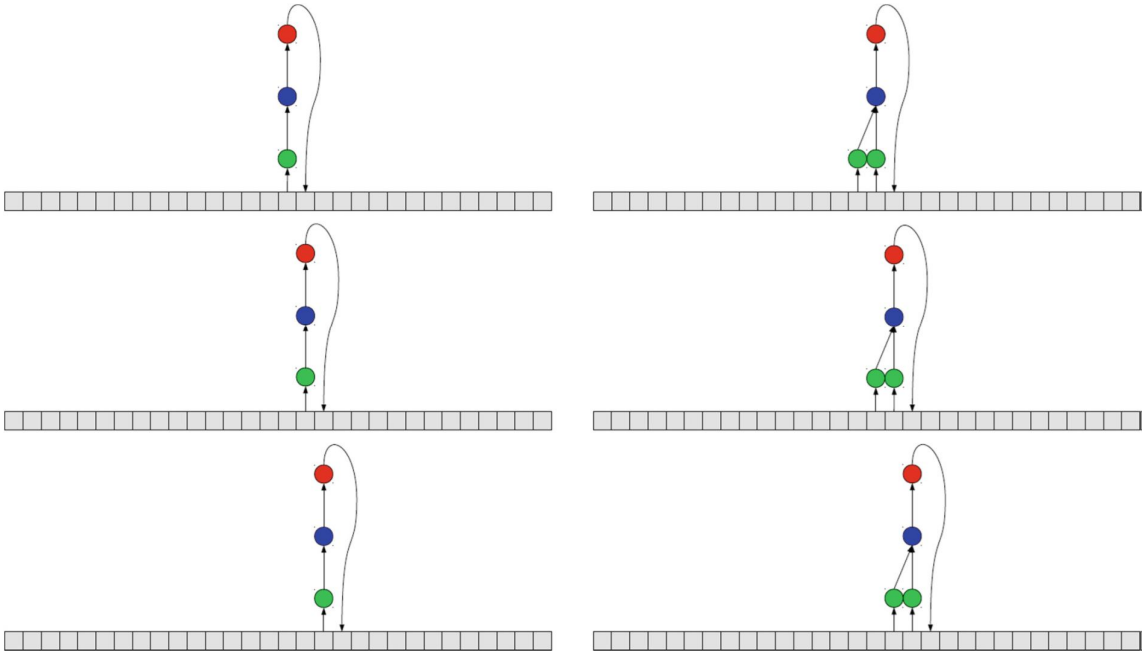
Когато времевите редове са твърде шумни, входящата информация може да бъде филтрирана с филтър Kalman например (Alexandrov, 2016).

В това проучване основната идея е, че вместо броя на скрити слоеве, увеличава се броят на невроните на входа и скритите слоеве се разширяват по време на обучението на невронна мрежа. Удължаването на входния слой е свързано с факта, че всеки времеви ред расте с поява на ново измерване. Целта на обучението е размерът на входния слой да бъде толкова голям, колкото размерът на пълната времева редица.

2.2.1 Предпоставки при моделирането

Условно времевите редове се разделят на минали и бъдещи. Стойностите, подадени във входа на изкуствената невронна мрежа, се наричат *lag* (закъснели) и са подмножество на миналите стойности, най-близки до бъдещите стойности. Стойностите, получени в изхода на изкуствената невронна мрежа, са прогнозата и те се сравняват с подмножеството на бъдещите стойности, наречени *lead* (водещи). Като основа на изкуствена невронна мрежа, за предложения модел се използва многослоен персептрон с входен, един скрит и изходен слой.

В предложения модел се използва набор от изкуствени невронни подмрежи и подмрежите се обединяват в обща изкуствена невронна мрежа. Най-малката изкуствена невронна подмрежа има 1-1-1 топология (фиг. 2.2.6 - вляво). Мрежата е обучена с примери, чийто вход има само една стойност. Целта в модела е прогноза за само една стойност напред във времето. Ето защо всички подмрежи имат само един изход. Фигура 2.2.6-вляво показва само 3 междинни примера на обучението. Всичките входни стойности се предоставят като примери за еластично обучение за обратно разпространение на грешката. Обучението спира на определено ниво на *epsilon* за пълна промяна на грешките на невронната мрежа.

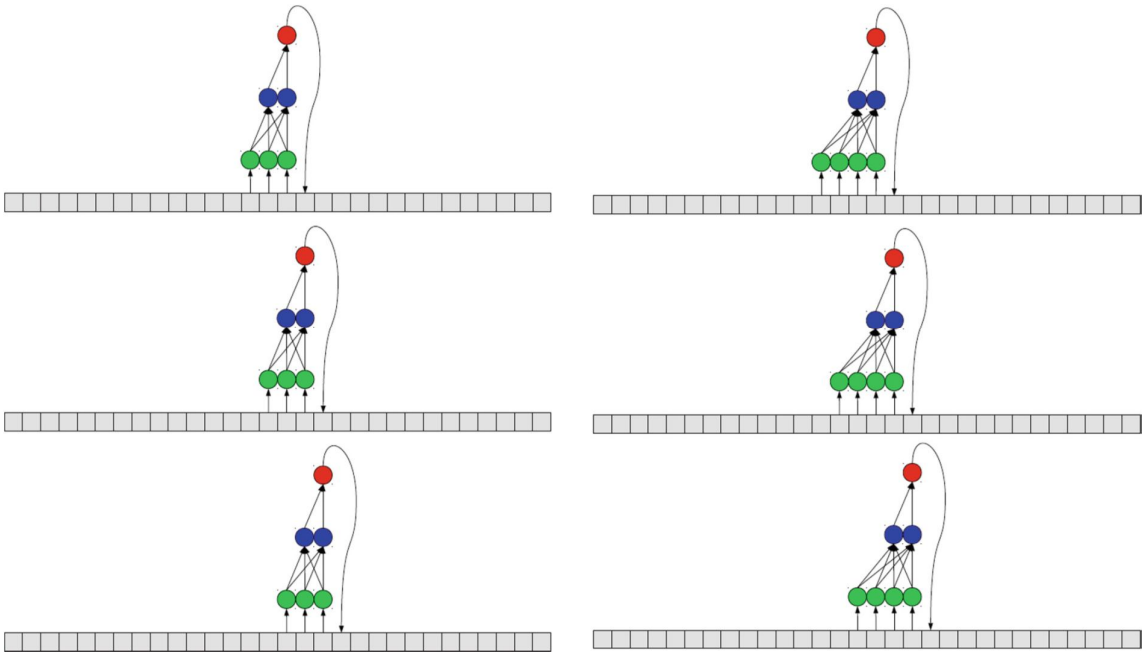


Фиг. 2.2.6. Обучение на изкуствени невронни подмрежи с 1-1-1 топология (вляво) и 2-1-1 топология (вдясно).

След обучение на 1-1-1 топология стойностите на теглата на първата подмрежа се зареждат във втората подмрежа с 2-1-1 топология (Фиг. 2.2.6-вдясно). Очевидно е, че една от стойностите на теглови коефициенти няма да бъде заредена, защото не е представена в първата подмрежа. Това тегло има стойността от предишното обучение на най-голямата подмрежа. Времевите редове са реорганизирани, за да предоставят две стойности за входни данни и да очакват една прогнозна стойност в резултата. След обучение на 1-1-1 топология стойностите на теглата на първата подмрежа се зареждат във втората подмрежа с 2-1-1 топология (Фиг. 2.2.6-вдясно). Обучението е същото както при първата подмрежа - обратно разпространение на грешката. Както при първата подмрежа, обучението спира на определено ниво на *epsilon* за пълна промяна на грешките на невронната мрежа.

Трета подмрежа има 3-2-1 топология. Размерът на скрития слой се избира автоматично чрез алгоритъм за постепенно подрязване, внедрен в Encog Machine Learning Framework (http://www.heatonresearch.com/encog/). Фигура 2.2.7-вляво показва два неврона в скрития слой, но това е само илюстративно - реалният размер на скрития слой се изчислява от алгоритъма. Алгоритъмът за обучение и критериите за спиране са същите като при предишните подмрежи.

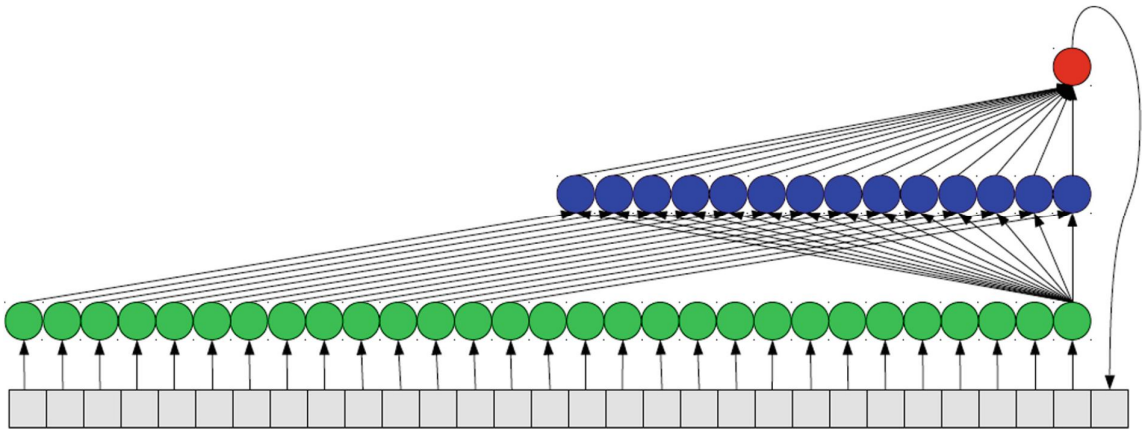
Четвърта подмрежа има 4-2-1 топология и отново размерът на скрития слой е само илюстративен (Фиг. 2.2.7-вдясно). Реалният размер на скрития слой се изчислява чрез алгоритъм на постепенно подрязване. Примерите за обучение са с по-малко в сравнение с предишната подмрежа, защото входният размер е с един по-голям. Критериите за обучение и спиране са същите като при предишните подмрежи. Фигури 2.2.6 и 2.2.7 показват само първоначалните 4 подмрежи. В изпълнението на модела са включени много повече подмрежи. Топологиите на подмрежите се формират чрез добавяне на един неврон във входния слой и коригиране на размера на скрития слой с алгоритъм за инкрементално подрязване. Крайната цел е да се достигне n-m-1 топология (фиг. 2.8), която обхваща всички известни стойности на времеви редове. Повечето връзки между входа и скритите слоеве на фиг. 2.8 са пропуснати за по-добра визуализация, но и двата слоя са напълно свързани в изпълнението на модела.



Фиг. 2.2.7. Обучение на изкуствени невронни подмрежи с топология 3-2-1 (вляво) и 4-2-1 топология (вдясно).

След обучението на най-голямата подмрежа йерархичната процедура се връща обратно в най-малката подмрежа. Стойностите на теглата от най-голямата подмрежа, които съответстват на връзките в по-малката подмрежа, се вземат и се зареждат в най-малката подмрежа. По подобен начин се вземат тегла от най-голямата подмрежа за останалите подмрежи в комбинация с теглата от предишните на по-малките подмрежи. Например, подмрежата с топология 4-2-1 ще вземе някои от своите тегла от 3-2-1

подмрежи, но връзките, които не са представени в по-малката подмрежа, ще бъдат взети от най-голямата подмрежа.



Фиг. 2.2.8. Обучение на изкуствена невронна подмрежа с n-m-1 топология. Някои от връзките между входния и скрития слой не се визуализират за по-добър външен вид

Общата идея зад предложения модел е постепенното обучение на състезателни по размер изкуствени невронни мрежи. Такова обучение е вдъхновено от естествените невронни системи, където биологичните клетки растат на брой и формират връзки помежду си. Често срещаният проблем при обучението по изкуствени невронни мрежи е размерът на мрежата. Чрез разделянето на най-голямата мрежа в много по-малки мрежи се постига ускоряване на учебния процес. В областта за прогнозиране на времеви редове е много добре известно, че най-старите измервания имат най-малко въздействие върху прогнозата. Предложеният модел взема предвид този факт и най-старите измервания се правят в най-голямата подмрежа, но те имат относително по-малко въздействие в крайната прогноза. Предложеният модел има по-висока степен на самоадаптация, тъй като когато се появи нова стойност във времевия ред, размерът на изкуствената невронна мрежа нараства, което означава, че фазата на обучение и фазата на работа са едновременни.

2.2.2 Експерименти върху изследването

Всички експерименти се правят чрез Java програма, където изкуствените невронни мрежи се изпълняват с API, предоставен от Encog Machine Learning

Framework (http://www.heatonresearch.com/encog/). Всички входни неврони нямат функция за активиране, тъй като тяхната задача е само да доставят входните сигнали в скрития слой. Невроните в скрития и изходния слой се използват с хиперболична функция. Предпочита се хиперболичният тангенс вместо сигмоидната функция, тъй като има симетрия според оста X. Тази симетрия помага за ускоряване на обучението, когато се използва обучение за обратно разпространение на грешката, тъй като изходните стойности на невроните имат положителни и отрицателни стойности. Със сигмоидната активационна функция изходът на невроните е само положителен, а отрицателните сигнали (ако са необходими) могат да бъдат постигнати само чрез отрицателни тегла.

Фиг. 2.2.9. Стойности на валутите за два месеца на дневна база - валутна двойка EUR / USD.

Като входни данни за експериментите се използват финансови времеви редове на FOREX пазар (фиг. 2.2.9 и 2.2.10). Данните се вземат от ежедневна двумесечна търговия за валутни двойки EUR / USD и USD / JPY. Стойностите на времевите редове се мащабират в диапазона от -0,99 до +0,99 с правилото за мащабиране MinMax. Изходът на изкуствената невронна мрежа се пренастройва обратно до първоначалния диапазон със същото правило, което се използва в обратна посока.

Резултатите от експериментите все още са в диапазона на статистическата грешка, която идва от сложността на финансовите процеси и високочестотния шум вътре в данните.

Фиг. 2.2.10. Валутни стойности за два месеца на дневна база - валутна двойка USD/JPY.

Тази хибридна структура на невронна мрежа може да се използва за прогнозиране на финансови времеви редове.

Предложеният модел за самонадграждащи се трислойни MLP за прогнозиране на времеви редове е обещаващ подход за ускоряване на обучението на изкуствени невронни мрежи. Нарастващият размер на входния слой включва максимална информация, налична във времевите редове, но предложената процедура за обучение на изкуствена невронна мрежа отчита, че по-старите стойности трябва да бъдат по-малко информативни.

2.3 Изводи

В тази глава се предлагат нови методи за анализ и прогнозиране на пазарни ценови движения чрез времеви редове и невронни мрежи.

В резултат на това са направени следните заключения:

1 В изследването до тук се обхващат основните аспекти от процеса по анализ – от дефиниране на проблема и поставяне на задачите, до представянето на методи за решаването им.

2 Във всеки един от етапите се извършва с представяне на реални доказателства, чрез които може да се идентифицира наличието на слабости или необходимост от намиране на по-рационален подход в разглежданата област.

3 Методите позволяват да се интегрират в системи за автоматизирана обработка и вземане на решения. Разработеният метод (MA Volatility Indicator) подобрява прецизността в осцилатор (Моментум) и работи в комбинацията от два инструмента EMA или SMA, като предлага нова методика за интерпретиране на резултатите при пазарни анализи и спомага за намаляване на риска от загуби и увеличаване на успех при автоматизираната търговия. Предложеният алгоритъм за обучение чрез самонадграждане в трислойни MLP ускорява обучението на ANN при прогнозиране на финансови времеви редове.

4 Методите представени до тук, може да бъде прилагани от специалисти в различни области в системи с прогнозен характер, за вземане на решения, анализиращи събития и процеси базирани на времеви редове.

Съдържанието на тази глава е отразено в публикациите:

1 Иван Благоев, Николай Докев, Комбиниране на Моментум с Един Метод за Прогнозиране на Пазарни Ценови Движения За По-Точни Резултати (Combination of Momentum with One Method for Forecasting of Market Trends to Improve the Results), Международна научна конференция “УНИТЕХ’17” – Габрово, 2017 *Selected papers*, ISSN 2603-378X, pp. II-265-II-270

2 Blagoev I., Improving the Momentum Oscillator Accuracy by a Method for Forecasting of Market Price Movements, Сборник доклади от межд. конференция, НВУ "Васил Левски", 14-15 юни 2018, Том 9, стр. 177-185. E-ISBN-13: 978-619-7246-20-9

3 Balabanov T.D., Blagoev I.I., Dineva K.I. (2018) Self Rising Tri Layers MLP for Time Series Forecasting. In: Vishnevskiy V., Kozyrev D. (eds) Distributed Computer and Communication Networks. DCCN 2018. Communications in Computer and Information Science, vol 919. Springer, Cham. https://doi.org/10.1007/978-3-319-99447-5_50 **SJR:0.188**

Глава 3. Решения за осигуряване на криптографска защита чрез приложение на времеви редове при криптографията и кибер сигурността

В области като статистика, обработка на сигнали, иконометрия и математическо финансиране се използват техники от времеви редове, за да се намери цикличност и да се предскажат бъдещи стойности. В представеното изследване се предлага този подход да се приложи към анализа на качеството на система за генериране на произволни числа (RNG) за осигуряване на криптографската защита в информационните системи. За нашето изследване ще предоставим числов масив, за да можем да анализираме продукцията от времеви редове от произволни стойности. Получените резултати се преобразуват от CSV данни в двумерен масив и се изобразяват в графичен вид. Предоставеният анализ може да помогне да се избегне предпоставка за атака на този генератор на произволни числа, като се предвиди следващата стойност.

3.1 Слабости в RSA чрез анализ с времеви редове, състояние на генераторите на случайни числа подпомагащи модулната криптография

Сигурността играе жизненоважна роля в областта на комуникационната система и Интернет. Стандартът за криптиране на данни (DES) и алгоритмите Rivest-Shamir-Adleman (RSA) са двата популярни алгоритма за криптиране, които гарантират поверителността и автентичността над несигурните комуникационни мрежи и интернет (Singh, 2013).

RSA е асиметричен алгоритъм за криптиране, който позволява на всеки да изпраща криптирани съобщения, които само притежателя на частния ключ може да декодира. Което е концепцията за асиметричната криптография. Този алгоритъм е съставен от трима математици в МИТ, от където идва и неговото име RSA (Rivest Rivest-Adi Shamir-Leonard Adleman). Като всяка една от трите букви на алгоритъма се включва в едно от имената на тримата му създатели. Принципът на работа може да се обясни накратко, като се генерира едно много голямо произволно число p, след това се генерира още едно такова число q и се изчислява тяхното произведение $x=p*q$, всъщност x е известен като публичен ключ. След това се използва този ключ за

извличането на друго също така много голямо число (частния ключ). Това изчисление е трудоемко, но иначе концепцията на алгоритъма като идея е сравнително проста.

3.1.1 Изследователите на (почти) секретния алгоритъм – слабости поради недостатъчна ентропия на RNG

Същността на RSA криптирането е, че използвайки само информацията, която публично се дава – публичен ключ, всеки може да кодира съобщение, което иска да изпрати на собственика на частния ключ. Но без да се знаят стойностите на p и q никой освен притежателя на частния ключ не може да декодира съобщението. И въпреки че всички знаят публичния ключ $x=p*q$ това не им дава никакъв ефективен начин да намерят стойности за p или q. Даже според група изследователи преди години се смяташе, че дори за откриването на 232-цифрено число, ще е нужно повече от 1500 години изчислително време (разпределени между стотици компютри), за да бъде компрометиран такъв частен ключ.

Така на повърхността, RSA криптирането изглежда неуязвимо. И можеше да се твърди така и до момента, но с изключение на един малък проблем. Почти всеки използва едни и същи генератори на случайни числа. За да се генерират прости числа, които изграждат криптографските ключове в RSA с високо качество е необходим наистина добър източник на ентропия. В конвенционалните компютърни системи, източниците на добра ентропия са сравнително оскъдни за такава задача. За това от години масово се ползват сийдове, които трябва да са произведени с качествена ентропия и след това изчисленията за новите RSA ключове се изпълняват през псевдо генератор на случайни числа.

Взимането на предвид този факт, може да се насочим към изследване от последно време според, което се заражда една нова идея и нека разгледаме отново общо познатия ни пример:

Да предположим, че Боб и Алис публикуват публични ключове онлайн. Но тъй като и двамата са използвали една и съща програма за генериране на произволни прости числа, има по-голяма вероятност публичните им ключове да имат общ основен фактор. Факторирането на публичните ключове на Боб или Алис поотделно би било почти невъзможно. Но намирането на общи фактори между тях е много по-лесно. Всъщност времето, необходимо за изчисляване на най-големия общ делител между две числа, е близко до пропорционално на броя на цифрите в двете числа. След като се идентифицира общият основен фактор между ключовете на Боб и Алис, може да се

фактурира, за да се получи основната факторизация на двата ключа. Така погледнато е възможно да се декодират всякакви съобщения, изпратени до Боб или Алис.

Въоръжени с тази идея, изследователи сканирали Интернет и започнали да събират публични ключове от съответния алгоритъм. За целта събрали 6,2 милиона реални публични ключа. След това са изчислили най-големия общ делител между двойки ключове, като се компрометира даден ключ всеки път, когато той споделя общ фактор с други ключове. В рамките на този експеримент са успели да разбият 12 934 RSA ключа. С други думи, ако се използва небрежно технологията и не се преодолеят описаните слабости, RSA криптирането осигурява по-малко от 99,8% сигурност.

На пръв поглед това изглежда като цялата история. Четенето от изследвания по въпроса (Ron was wrong, Whit is right) по-отблизо обаче се разкрива нещо по-обезпокоително. Според авторите, те са успели да извършат цялото изчисление за няколко часа върху актуална към момента едно-процесорна машина. Но погледнато чрез теоретичния фундамент на RSA, би трябвало да се предполага, че ще са необходими години, за да се изчисли GCD (най-голям общ делител) между 36 трилиона двойки ключове, а не часове, както е посочено според изследването.

Как са го направили? Авторите намекват в бележка под линия, че в основата на изчисляването им стои асимптотично бърз алгоритъм, който им позволява да сведат времето за изпълнение на изчисленията до почти линейно; но действителното описание на алгоритъма се пази в тайна от читателя, може би за да се предпази от злонамерена употреба. Само няколко месеца след публикуването на статията, последващите доклади вече обсъждаха подробно различни подходи, представящи бързи алгоритми (като това изследване: Quasi-linear GCD computation and factoring RSA moduli и дори се показва как да използвате графичните процесори, за да направите изчислението с груба сила по-бързо (Breaking weak 1024-bit RSA keys with CUDA https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=https://scholar.google.com/&httpsredir=1&article=1241&context=csse_fac).

Вероятно тук може да се каже, че не е добре да се споменават неща, ако те трябва да останат в тайна. Но от друга страна, ако слабостите в криптографски функции не се осветяват, рискуваме да се използват от злонамерени лица без това да е известно на останалите. В такъв случай, за да стигнем до резултатите от изследванията, трябва да се насочим към алгоритмите.

Имайки предвид характеристиките на криптографията и предлагания подход, то: Алгоритъмът ще се занимава с цели числа, имащи асимптотично голям брой цифри. Следователно няма да се разглеждат добавянето и умножението като операции с фиксирано и относително време.

За *n*-бита числа, се отнема *O(n)* време. Използвайки операция умножение, изглежда, че умножението отнема $O(n^2)$ време. Оказва се обаче, че има алгоритъм (Schönhage–Strassen algorithm), който работи във времето *O(n log2 n log log n)*.

Изчисляването на GCD с помощта на евклидовия алгоритъм ще отнеме *O (n2 log n log log n)* време. За пореден път обаче изследователите са намерили по-добър алгоритъм, работещ във времето *O(n log2 n log log n)*. За късмет, всички тези алгоритми вече са внедрени за нас в GMP (GNU MP Subquadratic), библиотеката на C++ с големи числа. За останалата част от изследването ще използваме нотация $\tilde{O}$, вариант на Big-O нотация, който игнорира логаритмични фактори. Например, докато изчислението на GCD отнема време *O (n log2 n log log n),* в нотация пишем, че отнема време $\tilde{O}(n)$.

3.1.2 Трансформиране на проблема

Определя се набора от публични RSA ключове с k_1.....k_n, където всеки ключ е произведение на две големи прости числа. Трябва да се обърне внимание, че n е общият брой ключове. Вместо да изчислим GCD на всяка двойка ключове, може да се изчисли за всеки ключ k_i GCD от него и произведението на всички останали ключове $\prod_{t=1} K_t$. Ако ключ k_i споделя един основен фактор с други ключове, тогава това ще даде основния фактор. Ако обаче и двата основни фактора на k_i се споделят с други ключове, изчислението няма да успее действително да извлече отделните първични фактори. Този случай може да е достатъчно рядък и да не си струва му да се отделя голямо внимание.

Алгоритъмът:

Алгоритъмът има леко необичайна рекурсивна структура, тъй като рекурсията се случва в средата на алгоритъма, а не в края.

В началото на алгоритъма всичко, което имаме, са ключовете,

k_1,

k_2,

k_3,....

Първата стъпка на алгоритъма е да свържете ключовете и да изчислите техните резултати,

$j_1 = k_1 . K_2$,

$j_2 = k_3 . K_4$,

$j_1 = k_5 . K_6$, ...

След това в рекурсия по последователността на числата j_1,.....,j_n се изчислява

$r_1 = GCD(j_1, \prod_{t \neq 1} j_t)$,

$r_2 = GCD(j_2, \prod_{t \neq 2} j_t)$,

$r_3 = GCD(j3, \prod_{t \neq 3} j_t)$, ……

Целта е да се изчисли $s_i = GCD(k_i, \prod_{t \neq i} k_t)$ за всеки k_j ключ. Важното тук е, че когато i е нечетно, s_i може да се изрази като

$s_i = GCD(k_i, r_{(i+1)/2}.k_{i+1})$

и че когато i е четно, s_i може да се изрази като

$s_i = GCD(k_i, r_{i/2}.k_{i-1})$

За да се разбере защо е така, може да се провери дали изразът от дясната страна на *GCD* е гарантиран, че е кратен на $s_i = GCD(k_i, \prod_{t \neq i} k_t)$, като същевременно е и делител на $\prod_{t \neq i} k_t$ Това от своя страна предполага, че изчислението на *GCD* ще бъде точно $GCD(k_i, \prod_{t \neq i} k_t)$, както се очаква.

Време за изпълнение: Нека с m се обозначат общия брой битове, необходими за записване k_1, k_2, ..., k_n. Всеки път, когато алгоритъмът се повтаря, се гарантира, че общият брой битове във влизане на рекурсията, е не повече от на предишното ниво на рекурсия; това е така, защото новите влизания са продукти от двойки елементи от стари такива.

Следователно всяко от нивата на *O(log n)* на рекурсия действа върху вход с обща големина на *O(m)* бита. Освен това аритметичните операции във всяко ниво на рекурсия отнемат общо най-много време *Õ(m)*. По този начин общото време на работа на алгоритъма също е *Õ(m)* (тъй като нивата на рекурсия *O(log n)* могат да бъдат усвоени в нотацията О-тилда).

Ако разгърнем времето за работа в стандартна Big-O нотация, получаваме

O(m log3 m log log m).

Практичен ли е подходът?

На пръв поглед, тройният логаритмичен фактор може да изглежда като изключващ използването на този алгоритъм. Но в друго изследване се оказва, че това представяне е доста разумно. В статията (Cloostermans, 2012) е установено, че алгоритъмът отнема време приблизително 7,65 секунди на хиляда ключа, което означава, че ще отнеме малко повече от 13 часа, за да се изпълнят 6,2 милиона ключа.

Също така се оказва, че един от LOG факторите може да бъде премахнат с помощта на друг подход, който изобщо избягва изчисленията на GCD, освен на първото ниво на рекурсия, за пример статията (Heninger, 2012). Така подобреният алгоритъм отнема около 4,5 секунди на хиляда ключа, което води до общо време на работа от около 7,5 часа за работа с 6,2 милиона ключа. Така че изчисляването, което би трябвало да отнеме години, се свежда до часове. И всичко, което е необходимо е прилагането на рекурсия, анализ с времеви редове, за да се използва слабостта в генерирането на случайните числа в системите.

В заключение може да се каже, че слабостите не изхождат от грешка в аритметиката на RSA. Те идват от технологичната слабост, с която се прилага RSA. Компютърните системи ако са от по-ново поколение имат хардуерни и софтуерни подобрения, които позволяват да генерират качествени случайни числа. Но въпреки това опасността от тази уязвимост остава. Понеже RSA се нуждае от наистина големи случайни числа. Съвременните критерии за надежден RSA ключ е минимум 2048 бита, като препоръчителната дължина е даже 4096 бита. При други изследвания също е установено, че между 4096, 8192 и 16384 бита RSA ключ, по-голямата сигурност на по-големите ключове е минимална. Причината също идва от ограниченията при генераторите на случайните числа. При по-големи RSA ключове са необходими

изключително големи истински случайни числа. Които в една компютърна система е крайно трудно да се получат. Дори да се използват силициевия модул за HwRng за целта, буфера на ентропия е 4096 бита и той се натрупва бавно, като ограниченията идват от технологията. При използването на RSA криптография, при системи значително по-оскъден хардуер като IoT, генерирането на RSA ключове ще е още по-слабо. Отново причините са същите, а този тип устройства често изобщо не притежават специализиран хардуер за обогатяване на ентропията. За това множество такива устройства се превръщат често в лесна жертва при кибер атаки.

3.2 Метод за оценка уязвимостта на случайните числа за криптографска защита в информационните системи чрез времеви редове

Както се казва, че в основата на всяка една криптираща система седи алгоритъм и генератор за случайните числа. За това се счита, че независимо, колко сложни криптиращи алгоритми се прилагат, те стават толкова уязвими, колкото е уязвим генераторът на случайни числа, който е в основата на тази система.

Ефективността на RNG се измерва чрез степента на ентропия за генериране на произволни числа.

Сложността на анализирането на даден генератор на произволни числа е функция на качеството на неговата ентропия, сезонност и тенденция към сблъсък. Това са моментите, когато генераторът на случайни числа ще генерира стойност, която е циклично или стойностно поле, което води до повторение или генериране на нова, но очаквана стойност. Чрез математиката на времевите редове е възможно да се определи ентропията във времето и е вероятно да се изчисли (прогнозира) евентуалното бъдещо повторно появяване на данните. Откриването на сезонност в получените стойности, отклонения или сблъсъци може също да показва слабости на генератора на случайни числа. Ако генераторът е качествен, тогава той ще следва анализ на много голямо количество статистически стойности от числов масив, генериран от него с висока степен на ентропия и непредсказуемост, което ще бъде много ресурсно интензивно и сложно. Това също ще направи много устойчив на атака в цялата криптография, свързана с този генератор.

3.2.1. Методи за генериране на произволни числа в PHP

Във връзка с написаното тук, нека разгледаме начини за генериране на произволни числа в технологията на езика за програмиране на PHP в различни операционни системи. Тук основното средство за случайни числа са PRNG:

1. Линеен конгресен генератор (LCG), напр. `lcg_value ()`

2. Алгоритъмът Marsenne-Twister, напр. `mt_rand ()`

3. Локално поддържана функция C, т.е. `rand ()`

Те се използват повторно и за функции като `array_rand ()` и `uniqid ()` в PHP. Недостатъкът на ентропията и генераторите на произволни числа на гореописаните функции се състои в лесното прогнозиране на бъдещите стойности на PRNG.

Недостатъчната ентропия произтича от факта, че първоначалните вътрешни състояния или SEED на PRNG са ограничени и изходът на стойности е в недостатъчен диапазон и това е предвидимо от лесно достъпните съвременни изчислителни ресурси.

Да разгледаме следния пример:

```
<?php
mt_srand(3231153718);
for ($i=1; $i < 15; $i++) {
    echo mt_rand(), PHP_EOL;
}
```

Изпълнявайки горния скрипт, получаваме следната последователност от 15 псевдослучайни числа:

```
818549686, 167964715, 1051297630, 679916735, 46602984, 1183523200,
591658099, 619786907, 50517956, 1766971703, 380190080, 1985369304,
1829774512, 859734050
```

Без значение колко пъти стартираме този скрипт, ще получим една и съща последователност от числа. Получихме стойността SEED = 3231153718, използвайки функцията `rand()`. В този случай, ако атакуващият получи стойността на SEED, използвана в Mersenne Twister PRNG на PHP, той / той ще може да прогнозира изхода

на mt_rand (). За това, като въпрос на защита, в този случай е от основно значение да се защити стойността на номера SEED.

За да получат стойност за SEED в PHP, разработчиците често получават стойност за SEED, като използват горния скрипт, или могат да използват mt_rand (), за да го дадат автоматично. Това е особено вероятно да бъде вярно, когато се използват по-стари PHP базирани системи и дори с по-високи версии на PHP, използвани в съвременните условия. Следователно съществува риск възстановяването на SEED от нападател. Въпреки пасивния си характер, това всъщност е истинска уязвимост. Дори информацията да бъде пропусната в определени точки в системата, тя позволява на нападателя да извърши допълнителни атаки, което е нарушение на принципа на задълбочена защита.

Нека си представим да работим с приложение, което използва следния изходен код за генериране на токен за различни цели на приложението:

```
$newtoken = hash('sha512', mt_rand());
```

Със сигурност има по-сложни начини за генериране на токен, но също така е хубав пример за само едно обаждане към mt_rand (), което е хеширано с SHA512. В действителност, ако програмист приеме, че функциите на случайните стойности на PHP са "достатъчно случайни", той ще бъде много по-склонен да вгради прост модел на използване (подобен на този, представен по-горе), стига да не включва думата криптография. SHA512 изпитва страхопочитание на програмистите, защото това е най-големият алгоритъм на стойност в семейството на SHA-2. Въпреки това, използваният по-горе метод за генериране на маркери страда от един недостатък - случайните стойности са ограничени до цифри (т.е. неговата несигурност или ентропия е близка до незначителна). Ако проверим продукцията на mt_getrandmax (), ще открием, че максималният произволен брой mt_rand () може да генерира само 2,147 милиарда. Този ограничен брой опции го прави уязвим за груба атака.

Ако имаме доста добра видеокарта от последните няколко поколения, нещата стават още по-лесни. В този случай приложението hashcat може да се използва като пример, като е един от софтуерните инструменти на грубата сила. Ако всичко е конфигурирано и работи правилно, hashcat ще намери стойността на хешираното произволно число в рамките на няколко минути. Това показва на практика как работи ентропията, която описваме тук. Функцията mt_rand() е ограничена до толкова

малко опции, че хешовете на SHA512 на всички възможни стойности могат да бъдат изчислени за кратко време. Следователно използването на хеш за скриване на изхода на `mt_rand()` е безполезно.

За да се защити този тип система, трябва да се генерират случайни стойности с по-високо качество. За използване в нетривиални задачи, PHP изисква източници на ентропия от висок клас, които могат да бъдат осигурени от операционната система. Linux обикновено се използва с `/dev/urandom`, освен ако не са инсталирани устройства с още по-висока ентропия.

PHP може да се осъществи чрез функциите за повторно използване на външната библиотека на OpenSSL, като се използва `openssl_pseudo_random_bytes ()` и `mcrypt_create_iv ()`. Тези две функции също са оптимизирани да използват криптографски защитен псевдослучайни генератор. В Windows източник на такава ентропия е CSPRNG, но засега PHP няма директен достъп до нея без специални разширения, които да я предоставят. Всъщност трябва да сме сигурни, че версията на PHP, която се използва на нашите сървъри, е активирала OpenSSL за Linux или Mcrypt за Windows.

В Linux, с правилната настройка, редовен генератор на произволни числа, който е от типа PRNG (който е псевдо генератор на произволни числа), често се зарежда от източник на висока ентропия `/dev/random`, което го прави устойчив на атаки. Обикновено се избягва директното четене от `/dev/random`, тъй като е с ограничен капацитет и може да доведе до блокиране на всички опити. Адресът към `/dev/random` се среща само в критични случаи, когато е необходимо.

3.2.2. Разбиране на RNG Entropy в Linux

Общи процедури, които предоставят случайни числа в Linux OS са:

1. `/dev/random` е истински генератор на случайни числа и се блокира, ако ви свърши ентропията. Този инструмент съхранява ентропията, събрана от системните параметри по време на работата му като достъп до диск, мрежов трафик, състояние на паметта, преместване на мишката и други прекъсвания на системата. Също така, ако процесорите на Intel използват, тогава се предлага допълнително предимство, ако

вашият системен процесор не е по-стар от произведения през 2013 година. Тази технология за сигурност на Intel е източник на ентропия на цифровия генератор на произволни числа, подобен на допълнителен източник на ентропия. Използването на `/dev/random` обаче трябва да се извършва внимателно, защото този източник на ентропия е ограничен и не можем да го използваме директно през цялото време.

2. `/dev/urandom` е генератор на псевдо произволни числа (PRNG) и той не се блокира поради изчерпването на ентропията. Може да се използва за рандомизиране на неограничен поток. Случайният поток се осигурява от PRNG структури и необходимите начални SEED стойности ще се презареждат периодично от /dev/random.

3. `/dev/hwrng` е допълнителен хардуер за истински случайни числа, който е специализиран и не е инсталиран в компютърните системи по подразбиране. Той осигурява шум от ентропия за поддържане на случайни числа.

Лесно е да се види колко ентропия е налична чрез следните команди:

```
$ cat /proc/sys/kernel/random/poolsize
4096
$ cat /proc/sys/kernel/random/entropy_avail
3868
```

където:

`/proc/sys/kernel/random/poolsize` се използва за деклариране на размера (в битове) на буфера Entropy Pool, например: Колко произволни числа трябва да съхраним, преди да спрем да „помпаме“ за повече.

`/proc/sys/kernel/random/entropy_avail` показва количеството (в битове) на текущо съхранени случайни числа в пула.

Сега можем да видим как използваме ентропията, за да бъдем по-ясни и след това да гледаме натрупването на ентропия в реално време:

```
$ watch -n 1 cat /proc/sys/kernel/random/entropy_avail
```

И сега можем да видим оценката на ентропията всяка секунда. Не правете нищо в началото и ще видите как стойността се увеличава бавно, ако изобщо. Например, в началото на това изречение има 2949 бита, но след това стигат 3327 бита, ентропията е била подпомогната от моето писане по клавиатурата и останалата системна активност.

Но как може да се използва част от тази ентропия сега? Лесен начин за това е да прочетете своят буфер с ентропия, като просто изхвърлите всичко, което е в /dev/random, генератора на произволни числа на ядрото:

```
$ hexdump /dev/random
0000000 d5c4 ff0a b8ef 9bdc ad95 480b e853 f0ef
0000010 e0cb 7c08 4bc4 daef 2b21 ea62 0eac 2c6c
0000020 d6bd 70e6 5d6f a7e3 0874 d52f 77df 6a2b
0000030 1909 efe8 9964 acee 2aad 2522 4ddb 1d0b
```

В друг прозорец може да се наблюдава наличието на ентропия, падащо до нулата, тъй като случайни стойности изтичат към терминала. С натисне на `Ctrl-C` се спира това безсмислено разхищение. Може би никога не трябва да се прави това на практика, освен с изследователска цел разбира се. Всъщност целият смисъл на събирането на ентропията беше в посяването на добър PRNG. След като това е направено, има твърде малко смисъл да се източва повече ентропия от буфера, освен ако нямате основание да смятате, че системата е компрометирана. Сега може да се види още един експеримент с този инструмент. С отваряне на браузър до някаква *https://* уеб страница и наблюдаване на наличната ентропия. Ентропията вероятно няма да падне, макар да се знае, че SSL връзките ще изисква генериране на секретни произволни числа. Това е така, защото всеки съвременен браузър използва `/dev/urandom` вместо `/dev/random` за своите случайни числа.

Но понякога системите имат проблеми с натрупването на ентропия в буфера и резултатът изглежда смущаващ:

```
$ cat /proc/sys/kernel/random/entropy_avail
96
```

От представения пример машината произведе ентропиен резултат от 96 бита и увеличаването на тази стойност е твърде бавно и недостатъчно. Това е много лошо за всяка отговорната система и нейните средства за гарантиране на сигурността, които изискват добри случайни числа, особено като криптографията. Причината за това се крие в инсталирането на ОС на виртуален хост, а не директно на хардуерът. Понякога виртуалните машини и техните драйвери за виртуализация са източникът на този проблем. Ако хардуерните драйвери работят правилно, вашето ядро лесно ще осигури

повече от 3000 бита ентропия в буфера. Но за всеки случай мисля, че е необходимо да проверите как вашата система натрупва ентропия по всяко време с нова инсталация на ОС. Едно от възможните решения да се контролира това и да стартира специализиран софтуер подпомагащ събирането на случайни числа. Това е демон, който използва събития, които могат да се считат за сравнително случайни при работата на машината, за да се произведат повече и по-качествени случайни числа. Например процесорното „трептене“, състоянието на паметта, входно изходни операции, мрежов трафик могат да добавят още ентропия към буфера на системата. Инсталирането и основната настройка са следните:

```
# apt install haveged
# systemctl start rngd
# update-rc.d haveged defaults
# rngd -r /dev/urandom
```

На система със сравнително умерен трафик:

```
# pv /dev/random > /dev/null
  40 B 0:00:15 [   0 B/s] [                    <=>              ]
  52 B 0:00:23 [   0 B/s] [                           <=>       ]
  58 B 0:00:25 [5.81 B/s] [                              <=>    ]
  64 B 0:00:30 [6.05 B/s] [                                 <=> ]
^C
# systemctl start haveged
# pv /dev/random > /dev/null
7.12MiB 0:00:05 [1.43MiB/s] [               <=>               ]
15.7MiB 0:00:11 [1.44MiB/s] [                       <=>         ]
27.2MiB 0:00:19 [1.46MiB/s] [                           <=>     ]
  43MiB 0:00:30 [1.47MiB/s] [                                <=> ]
^C
```

С помощта на командата `pv` може да се види колко данни се предават за целта. От показания поток на данните се вижда, че преди `haveged` се получаваха 2.1 бита в секунда (B / s), докато след това се получават ~ 1.5 MB / sec.

3.2.3. Времеви редове за генератори на случайни числа

Очевидно е, че темата за RNG и PRNG заслужава изключително внимание, особено защото цялата криптографската защита в информационните системи разчита на нея. За това е изключително важно да сме сигурни в качеството на системата или механизма за генериране на произволни числа, техните желани характеристики като псевдослучайност, сложност и чувствителност към промени в параметрите (Teh, 2020). Ето защо е желателно да се оцени такъв хардуерен или софтуерен генератор с повече от един метод за анализ. Традиционните мерки за RNG, които се използват за количествено определяне на тяхното функциониране, са предимно обобщени статистически данни, отнасящи се до отклонения от математическата случайност (Oomens, 2015).

В изложеното изследване като средство за оценка на надеждността се разчита на възможностите на времевите редове. Спецификата на RNG и PRNG позволява това, тъй като улавянето на потока на изходните числови стойности е последователност и само по себе си е подредено във времето. Такъв поток от числови стойности може да бъде описан, както следва:

$$N = T * V$$

където:

N - дължината на числовия ред,

T - време (продължителност) на генерирането на числа,

V - брой генерирани числа за единица време.

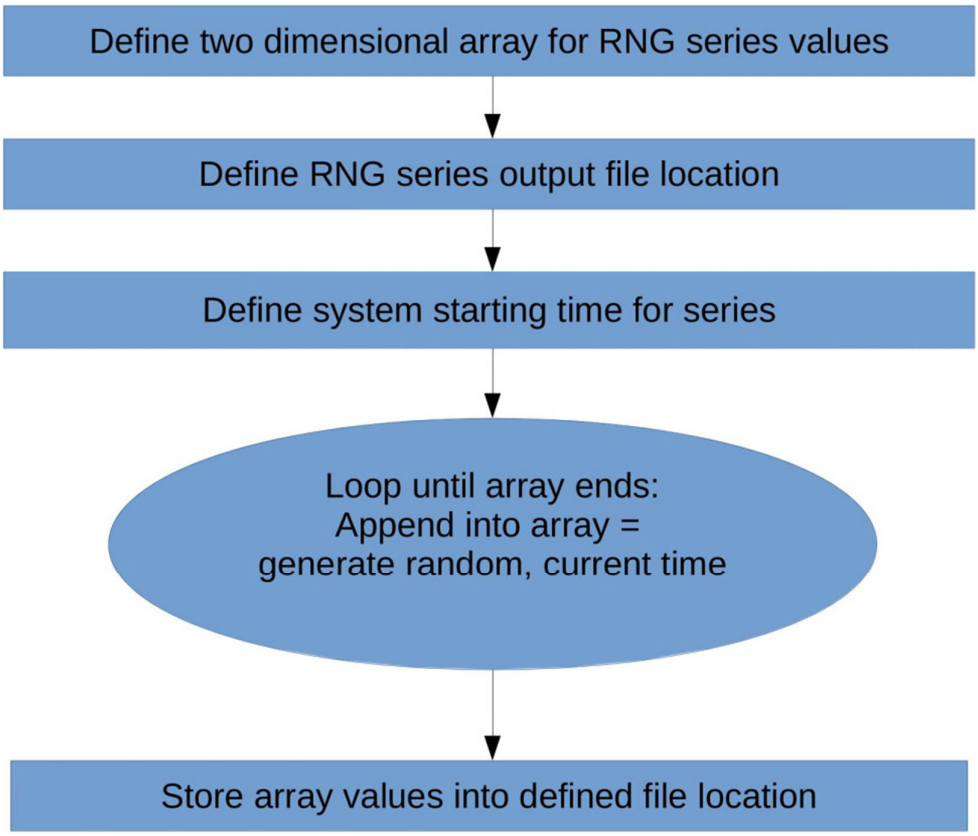
Следователно, потокът от цифрови данни, който се получава ($N = T * V$), е времеви ред. Чрез математиката на времевите редове е възможно да се определи ентропията във времето. Ако един генератор не е много надежден, неговите слабости биха могли да се намерят за по-кратък масив от генерирани числа, за които ще са необходими по-малко ресурси за обработка и анализ.

За текущото изследване ще се използва числов масив, който няма да бъде създаден от висококачествен генератор на случайни числа, а от посредствен такъв. Идеята е да се анализира времевата линия на изхода от случайни стойности от среден клас компютърна система, с каквато най-често всеки разполага. Целта е да се проследи подходът, а в последствие и с по-качествен генератор на произволни числа също може

да се приложи. Но за анализът на по-качествените случайни числа, трябва да се разполага с повече изчислителна мощност и значително по-голям обем от генерирани произволни изходни данни във времевият ред.

3.2.4. Проучване на генератори на случайни числа с времеви редове

Предполага, се че времевите редове като стохастичен процес могат да бъдат използвани за анализ и на RNG. За тази цел е разработен алгоритъмът за откриване на повтарящи се модели (patterns) от данни в генерираните от RNG времеви редове, който подробно е обяснен в следващо изложение.



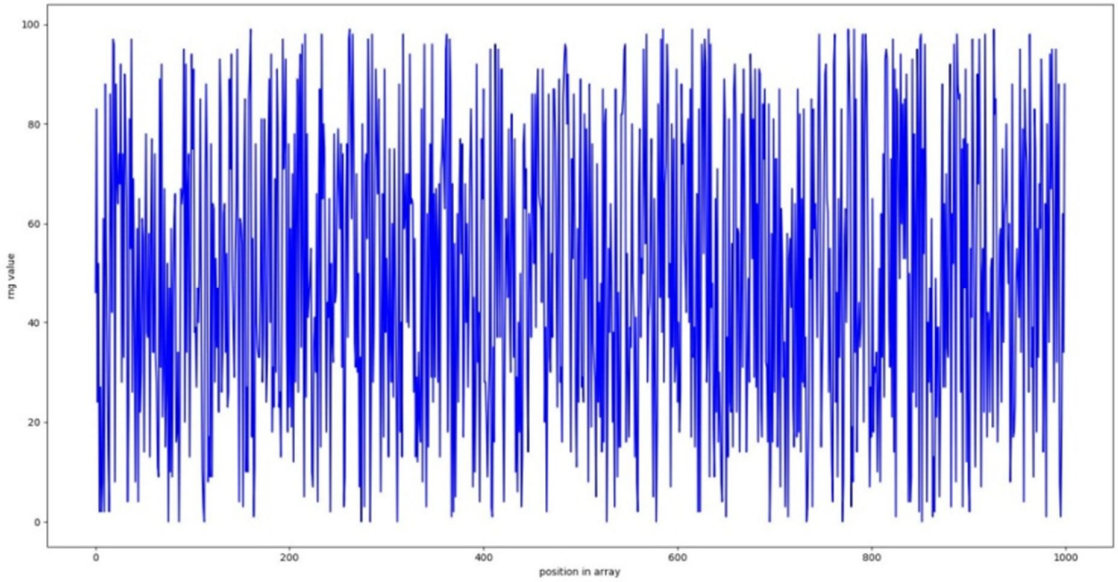
Фиг. 3.1. Експортиране и запазване на сурови файлове с данни от дефиниран RNG източник.

На фиг.3.1 е представена блок схема за генериране на стойности на RNG от избран източник в сериен масив.

За нуждите на това изследване компютърната програма за генериране и запис на времеви редове от RNG е използван езикът C # и платформата Mono с отворен код работеща под Linux с целевата рамка .NET Framework 4.7, тя е еквивалентна в Windows на Microsoft .NET Framework 4.7. Така че компилираният пренос на програма или програмен код е съвместим и с двете операционни системи.

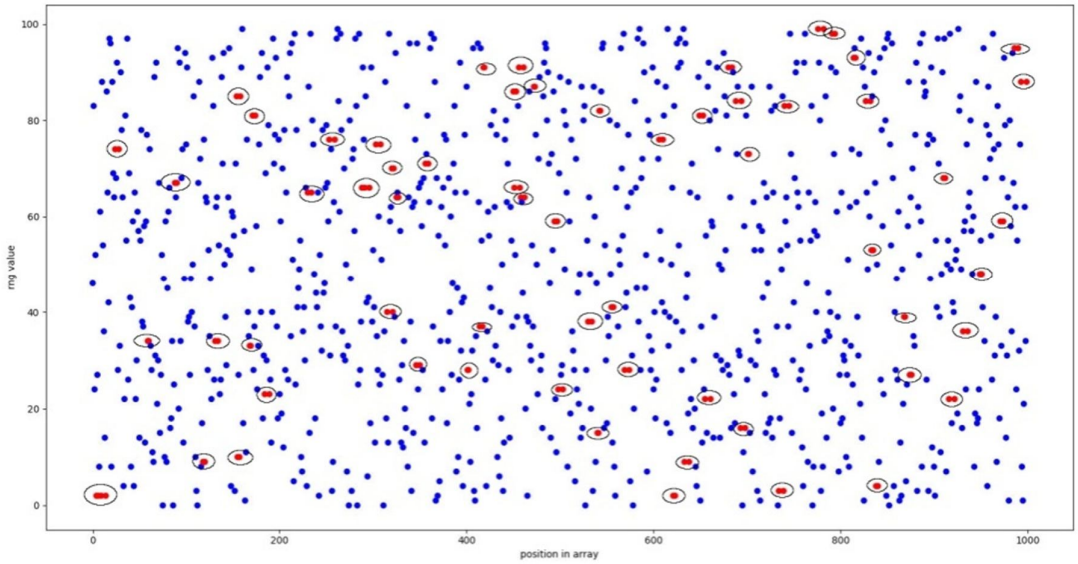
Логиката на програмата е следната: генерират се в цикъл случайни числа чрез System.Random, пренасочва се изхода и се записва в редица от времеви ред, като се получава няколко MB файл от тип CSV. Размерът на файлът зависи от степента на ентропия на RNG, който е целта на проучването. По тази причина размерът на файла е параметризиран и ще се дефинира и ще може да се определя според желаните нужди. Ако анализът покаже, че не се намират елементи на предсказуемост, като например слаба ентропия (Wollstadt, 2014), колизии или сезонност, ще се наложи да се увеличи размерът на изходните данни, за да се натрупа достатъчно статистическа информация, докато се намери пределът на недостатък на изследвания генератор на случайни числа.

Събраните данни със случайни числа се представят графично, което помага за по-лесно забелязване на важните елементи от времевите редове (фиг. 3.2):



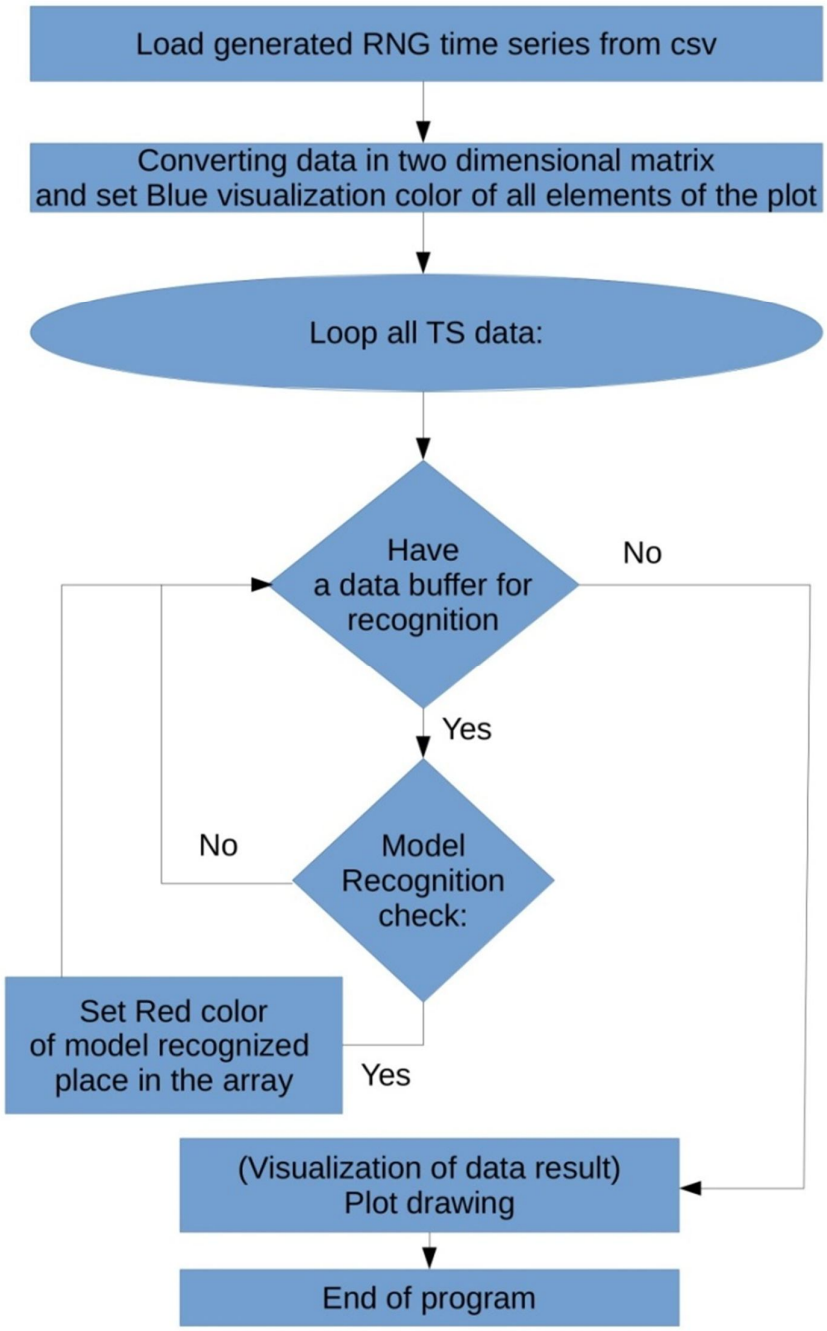
Фиг. 3.2. Визуализиране на данните получени от System.Random, като линия на шум.

На пръв поглед с резултатите от данните от System.Random на фиг. 3.2 всичко е наред и е възможно да се мисли, че имат добро качество на ентропия. Но нека предложа още един начин в друг графичен изглед, за да се уверим в преценката си.



Фиг. 3.3. Представяне на данните от System.Random във визуализация тип поле с точки

Представянето на едни и същи данни с различна графична интерпретация, като поле с точки от стойности на диаграма може да помогне за разкриването на някои проблеми с качеството на изследваните стойности от System.Random (фиг. 3.3). Като критични места може да се определят случайните стойности, които периодично се повтарят една след друга в масива. Те са маркирани с червени точки и са заобиколени от кръг. Разбира се, тази графика се генерира от компютърна програма (фиг. 3.4), която в случая е написана на Python, за да послужи за нуждите на това изследване. Както и след човешки анализ на данните, позволява да се определи повтарящият се модел, който да се използва по-късно за автоматизирано прогнозиране на стойности при работа с System.Random в друга компютърна програма. От получените резултати и на двете фигури е видно, че има случай при които в сравнително кратки периоди от време и дори последователни, периодично се получават еднаквите стойности. Което би било обезпокоително ако такъв генератор се използва за криптография и позволява да се атакува чрез прогнозни стойности.



Фиг.3.4. Блок-схема за откриване на повтарящи се модели.

В този случай изследваните данните от System.Random, са само няколко MB. Но ако се изследва по-качествен генератор на случайни числа, този размер значително ще нарасне. Тоест може големината на данните от времевия ред и изчислителната мощ за тяхното изследване са зависими от качеството на ентропия на генератора на случайни

числа. След това, ако анализът покаже, че в получените данни нямат прогнозни елементи като лоша ентропия, сблъсъци, прогнозни модели или сезонност, ще е необходимо да се увеличи техният размер, за да се натрупа достатъчно статистическа информация, която да покаже недостатъците на съответния генератор на произволни числа.

Полученият необработен файл от генератора в нашия случай работеше с програма, която създадохме на Python версия 3. Кодът на тази програма е заимстван от предишната и е усъвършенстван за по-добра визуализация. Логиката му е описана на фиг.3.4. Лесно е да се визуализират данните в линейна графична версия с двуизмерна координатна система. Но както вече се убедихме, че графиката прилича на общ шум, генериран от случайни числа, не може да бъде единственият начин за анализ, на който разчитаме. За това в следващата интерпретация на същите данни в графиката като точки от стойности на графиката проблемите вече са маркирани (фиг. 3.3).

Визуализацията на фиг. 3.3 показва слабостите на обработените резултати. Моделите на повторното появяване се срещат периодично във времето. Тези случаи са оцветени в червено от нашата програма, като се използват предварително определени модели за прогнозиране, както беше споменато по-рано. В конкретния случай на разглежданите времеви редове, тези случаи са 65 случая от масив с 1000 стойности. Може да се каже, че 6,5% прогнозни числа от генерирания RNG масив са значителен резултат. Обикновено последователност от псевдослучайни числа се инициира от SEED вътре в PRNG (Koeune, 2005). Ако такъв генератор на произволни числа се използва в криптографията, произведените от него SEED стойности могат да бъдат атакувани успешно. Чрез прогнозиране на следваща стойност SEED или чрез наблюдение на предадени криптирани данни, стойностите в основата на системата за криптиране могат да бъдат възприети в определен момент.

В други случаи, ако един атакуващ има повече информация за дадена система или за кодираните данни, той може да успее да преодолее криптографската защита и дори по-бързо. Намирането на повтарящи се модели също позволява автоматизирано търсене по програма и прогнозиране на следващите PRNG стойности.

3.3 Пренебрегнати рискове в киберсигурност в доставчиците на услуги за публичен Интернет хостинг

До тук изследването за анализи на качеството на случайните и областите в които се срещат проблеми успя да засегне криптографските алгоритми, езици за програмиране и операционни системи. Сега фокусът се измества върху масово предлагани публични хостинг услуги. Отново следва да се използва анализ на числа от времеви ред, които ще са продукт от генератор на случайни числа. Но темата е не по-малко чувствителна, защото засяга масови услуги, които често се използват и тяхната кибер сигурност. В ерата на масовата дигитална трансформация темата за кибер сигурността е от изключително значение. В крайна сметка и нормалното технологично развитие на човечеството щеше да доведе до все по-масова дигитална трансформация. Защото ползите за обществото и икономическото развитие са от изключително значение за света. Все повече дейности и процеси стават по-продуктивни и ефективно управлявани чрез технологии. Дигитална трансформация набида скорост и навлиза все по-бързо. На пръв поглед изглежда, че човечеството е много по-подготвено за това технологично предизвикателство от колкото сме предполагали. До известна степен това е така, но броят на кибер престъпленията се е увеличил и посегателството върху лични данни, парични средства и информация ескалира до невиждани до момента нива. Всичко това е силен индикатор, че от страна на кибер сигурността има много повече за наваксване в този преход. За това целта на проучването е да се обърне вниманието на проблемите в кибер сигурността в публичните хостинг услуги, които са лесно достъпни и представляват голям дял от масовото потребление.

За нуждите на настоящото изследване е използван уеб хостинг доставчик, който е един от популярните в бранша. Услугата за уеб приложения е инсталирана на масово предлаган нает споделен хостинг. Добавен е уеб сертификат и SSL достъпът е активиран, като всичко работи на стандартните портове за комуникация. Контролът над услугата не позволява да се променя конфигурацията на криптографските шифри, от която зависи цялото криптиране, което касае протоколът за криптирано тунелно свързване TLS. За да бъде надеждно защитена една уеб услуга, тя силно разчита на протокола TLS. Който може да бъде различни версии и да съдържа различен набор от криптографски шифри, много от които са вече остарели и компрометирани. За това, за да разчитаме на защитата на надеждна криптографска свързаност, настройката на

работата на този протокол е изключително важна. След извършеното сканиране на протокола може да се види предлагания списъкът с TLS версии и криптографските шифри, които са конфигурирани в предоставената услуга:

```
1. Protocol version: TLSv1.0:

- cryptography algorithm ciphers:

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,  (*) – A->B (till February 2020)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (*) – A security class → B
security class (till February 2020)

- compression: not supported

2. Protocol version: TLSv1.1:

- cryptography algorithm ciphers:

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (*) – A security class → B
security class (till February 2020)

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (*) – A security class → B
security class (till February 2020)

- compression: not supported

3. Protocol version: TLSv1.2:

- cryptography algorithm ciphers:

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (*) – A security class

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (*) – A security class

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (*) – A security class

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (*) – A security class

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (*) – A security class

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (*) - A security class

- compression: not supported

Elliptic curves for Diffie Hellman maintained by the server in a
preferential order - secp256r1, secp521r1, brainpoolP512r1,
brainpoolP384r1, secp384r1, brainpoolP256r1, secp256k1, sect571r1,
sect571k1, sect409k1, sect409r1, sect283k1, sect283r1
```

От представения списък, може да се види, че криптографските протоколи, които сървърът предлага: TLSv1.0 и TLSv1.1 изобщо не трябва да се поддържат и предлагат, тъй като имат отдавна известни слабости и са остарели. От гледна точка на кибер сигурността, възможността за установяване на връзка между клиентски сървър чрез тях е значителна слабост.

Протоколът: TLSv1.2 е все още актуален и се продължава да е одобрен за използване, но не в пълния му вид. Във включените криптографски алгоритми и редът им се срещат слаби и надеждни алгоритми. [6] За установяване на тунелна свързаност списъкът на алгоритмите, предоставен от този протокол, трябва да бъде редуциран само до надеждните такива, т.е. до следния актуален към момента вид:

• TLSv1.2 и идентификатори на поддържаните текущи алгоритми за криптографски шифри:

TLS_ECDHE_RSA(ECDSA)_WITH_AES_256_GCM_SHA384 (secp521r1, secp384r1) – A security class

TLS_ECDHE_RSA(ECDSA)_WITH_AES_128_GCM_SHA256 (secp521r1, secp384r1) – A security class

TLS_ECDHE_RSA(ECDSA)_WITH_AES_256_CBC_SHA384 (secp521r1, secp384r1) – A security class

TLS_ECDHE_RSA(ECDSA)_WITH_AES_128_CBC_SHA256 (secp521r1, secp384r1) – A security class

• за асиметрична криптография с RSA ключът не трябва да е по-малък от RSA4096 (ECDSA384 +)

• компресия: не се поддържа (разрешаването на компресия също отваря уязвимости в криптографията).

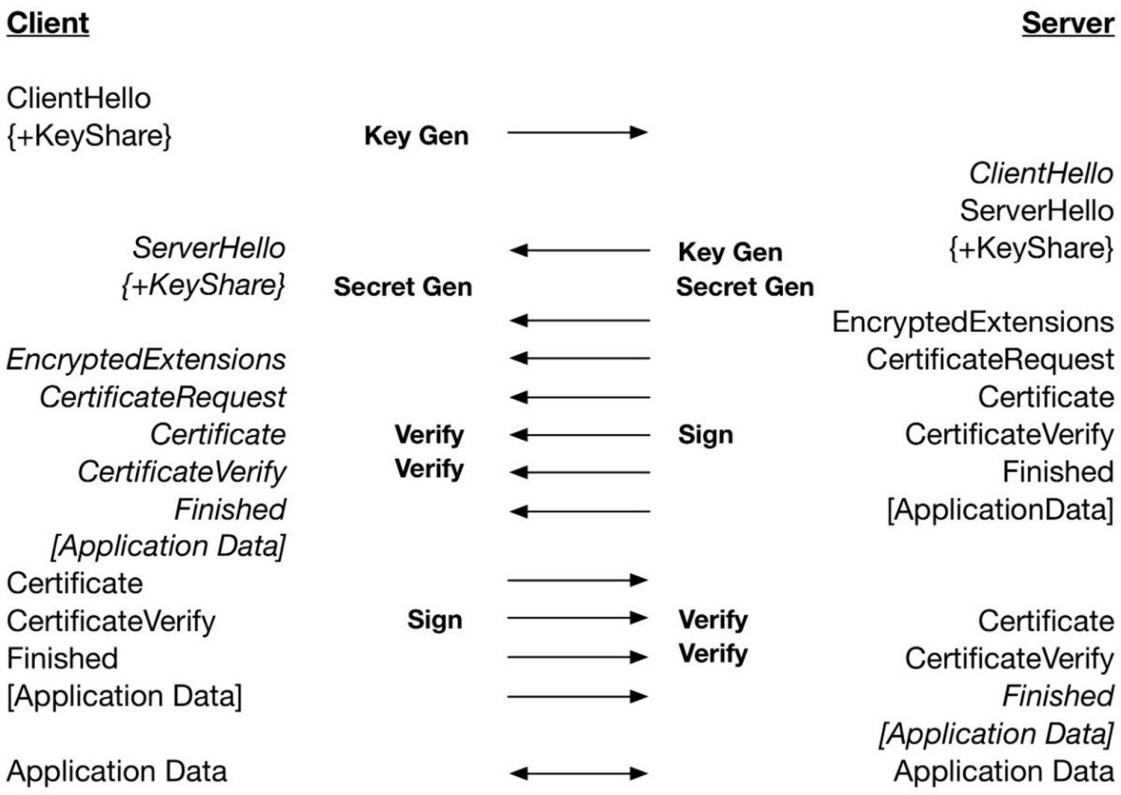
Друг съществен недостатък е, че протоколът TLSv1.3 не се поддържа, това за момента е най-сигурният актуален протокол от семейството на TLS за тунелна свързаност. В него вече липсват всички уязвими криптографски шифри и компрометирани криптографски алгоритми. Също така има някои много фундаментални промени в начина на установяване на свързаността, които значително увеличават кибер защитата на потребителите и сървъра. Освен това, когато потребителите се удостоверяване с клиентски сертификат, личните данни и информацията за сертификата е скрита за мрежовите шпиони. В предишните версии на протокола TLS беше възможно при подслушване да се събира информация за потребителите, които са се удостоверили със сертификати пред сървъра.

Но да се върнем на всеизвестния факт, че цялата криптографска защита е силна колко силен е генераторът на случайни числа в нейната основа. За това изследването се премества отново върху качеството на RNG ентропията, което подсигурява хостинг услуга. Но трябва да се има в предвид, че тази услуга е споделен уеб хостинг. Особеното е, че платформите от този тип споделят целия хардуерен ресурс между голям брой потребители и техните приложения за уеб услуги. Все още не се знае дали

този споделен хардуер има истински RNG. Но ако на сървърът съществува True RNG и много потребители чрез своите уеб приложения едновременно „източват“ този споделен RNG ресурс, тогава може да се предполага, че RNG ентропията може да се срине. Следователно цялата RNG ентропия ще бъде компрометирана, но този факт ще остане невидим за обикновения потребител на споделения хостинг. Това за злонамерен кибер престъпник с достатъчно познания, ще бъде идеалният момент за всякакви атаки, заради появилия се пробив в криптографска защита. Следва изследване върху състоянието на услугата, която включва:

- IP address;
- TCP ports;
- Web service;
- cPanel service;
- DNS administration panel;
- Mail service;

За са защитени при предаване данните по HTTP, каналът между крайния клиент (обикновено уеб браузър) и сървъра се криптора от TLS тунелен протокол, който на ниво пакет обгръща предаваните в чиста форма данни. TLS се състои от 2 фази - ръкостискане и установяване на канал чрез сесиен ключ. Нивото на информационна сигурност се определя от договорения набор от криптографски алгоритми, дължини на ключове и генерирани произволни числа, които се обменят между сървър и клиент във фазата на ръкостискане (Фиг. 3.5):



Фиг. 3.5. TLSv1.2 диаграма за ръкостискане (handshake)

Непредсказуемостта при получаване на случайни числа от системата, значително влияе върху първоначалните резултати от криптографските операции. Ключът на сесията TLS се формира след трансформации с произволни числа, генерирани от сървъра и клиента. Тъй като клиентът не винаги разполага с надежден метод за получаване на реални случайни числа, тази задача по подразбиране винаги се прехвърля от страна на сървъра. Ето пример за Hello ръкостискане и получаване на случайно число от сървър:

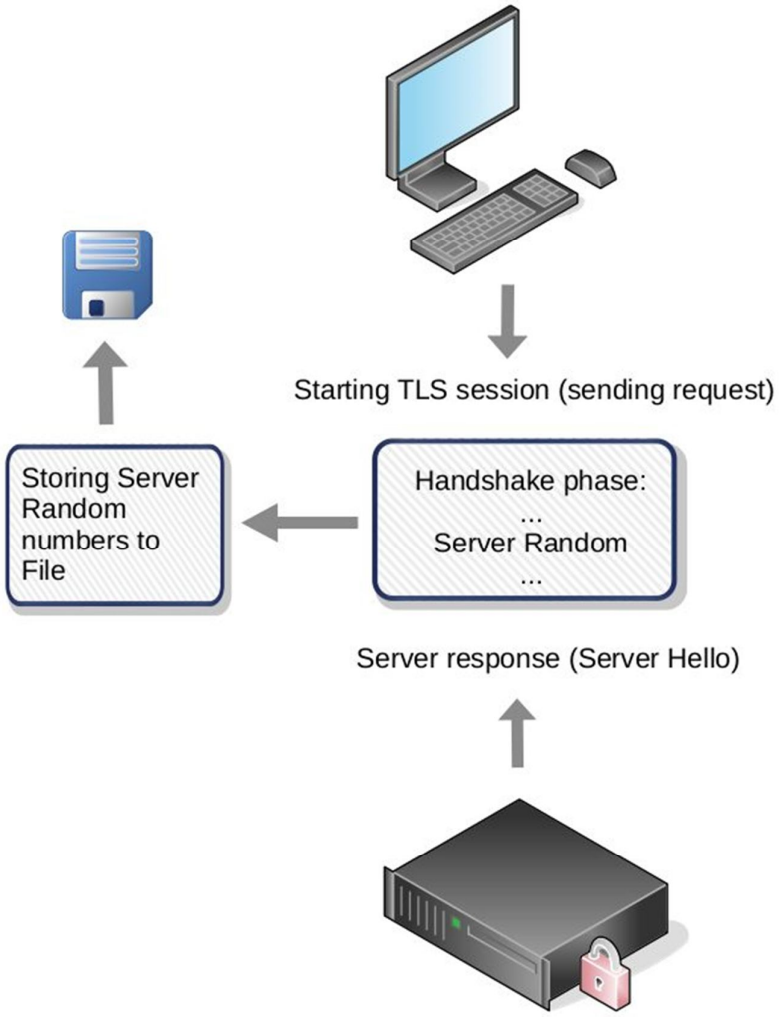
```
Version: 3.3 (TLS/1.2)

SessionID: 23 6C F0 EA 52 FA 7A E9 40 35 AA 23 17 55 1E 24 6C 9D C8
81 59 F5 CF 92 30 D2 11 1D 12 F9 2A 33

Random:          95 C6 63 18 AA 32 44 47 28 00 4B 94 2D AA F9 3B 12
9D 69 54 4B 45 1A B1 1E CA 4D DE B0 A3 86 5F

Cipher:          TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 [0xC02F]
```

```
CompressionSuite:     NO_COMPRESSION [0x00]
Extensions:
           server_name      empty
           renegotiation_info    00
           ec_point_formats uncompressed [0x0],
ansiX962_compressed_prime [0x1], ansiX962_compressed_char2 [0x2]
           ALPN       http/1.1
```

Представения кадър ни подсказва, от къде може да се извадят необходимите случайни числа, за да се измери качеството на ентропията. Необходимо е да се извлече масивът от случайни числа (който е показан в кадъра по-горе) и идва от източник, отговорен за работата на криптографията. В конкретния случай може да се направи по следните два начина:

1. Създаване на програмен код, който може да бъде инсталиран в предоставеното ни уеб хостинг пространство и изпълнен, като се извика уеб клиент. Получените произволни числа могат да бъдат записани във файл или изпратени директно чрез поток на клиента.

2. Дори и да нямате контрол върху хостинга, това може да се направи като клиентът при отправяне на уеб заявки към хостинг услугата. С помощта отново на написана на Python компютърна програма, която установява TLS връзка като клиент. След това във фазата Server Hello и TLS ръкостискането се извличат данните от променливата Random и се записват във файл. Когато тази операция се повтори достъчно, ще се съберат данните необходими за изследване на качеството на произволните числа. Ако се ползва този подход обаче, трябва да се предвиди известно закъснение между сесиите, за да не се прекали с отварянето на прекомерен брой връзки към сървъра и може да затрудни неговата работа. Което обаче ще доведе до по-бавно събиране на данните спрямо подход 1, но пък ще е също е безопасно. Фигура 3.6 визуално описва този процес.



Фиг. 3.6 Схема за изцличане на генерирани случайни числа чрез началната фаза на криптографска тунелна свързаност

Следвайки един от горните два метода се събира двоичен масив, който под формата на данни се записва във файл. Тези данни със случайни числа могат да бъдат подложени на изчислителен анализ с висока интензивност, който ще оцени качеството на ентропията им. В нашето изследване е използван специализираният софтуер с отворен код Dieharder на Робърт Г. Браун от Физическия факултет на университета Дюк (Brown, 2021). Резултатите от анализа са в **Приложение 1.**

Следва кратко описание на математическия анализ, приложен за определяне на качеството на масива от случайни числа чрез този инструмент:

Birthday spacing's: Изберат се m рождени дни в година от *n* дни. Броят се интервалите между рождените дни. Ако *j* е броят на стойностите, които се срещат повече от веднъж в този списък, тогава *j* е асимптотично разпределен по Поасон със средно m3 ÷ (4n). Опитът показва, че n трябва да е доста голям, да речем $n \geq 218$, за сравняване на резултатите с разпределението на Поасон с тази средна стойност. Този тест използва $n = 224$ и $m = 29$, така че основното разпределение за j се приема за Поасон с $\lambda = 227 \div 226 = 2$. Взема се проба от 500 *js* и се получава X2 test тест за добро състояние и се осигурява стойността *p*. Първият тест използва битове 1–24 (броене отляво) от цели числа в определен файл. След това файлът се затваря и отваря отново. После битовете 2–25 се използват за определяне на рождени дни, следва 3–26 и така нататък за битове 9–32. Всеки набор от битове предоставя p-стойност, а девет *p*-стойности предоставят резултатът за Kolmogorov–Smirnov теста.

The overlapping 5-permutation test: Това е тестът OPERM5. Той разглежда последователност от един милион 32-битови произволни цели числа. Всеки набор от пет последователни цели числа може да бъде в едно от 120 състояния, за 5! възможни подреждания на петте числа. По този начин 5-то, 6-то, 7-мо, ... редицата числа се намира в някакво състояние. Тъй като се наблюдават много хиляди преходи на състоянието, се правят кумулативни преброявания на броя на появите на всяко състояние. Тогава квадратичната форма в слабата обратна на ковариационната матрица 120 × 120 дава тест, еквивалентен на теста за съотношението на вероятността, че броят на клетките от 120 идва от определеното (асимптотично) нормално разпределение с посочената ковариационна матрица 120 × 120 (с ранг 99). Тази версия използва 1000000 цели числа, два пъти. Този тест може да има неразрешени грешки, водещи до постоянно лоши p-стойности;

Ranks of matrices: Това е BINARY RANK TEST за матрици 32x32. Формира се случайна двоична матрица 32x32, всеки ред 32-битово произволно цяло число и се определя рангът. Този ранг може да бъде от 0 до 32, ранговете по-малко от 29 са редки и броят им се обединява с тези за ранг 29. Ранговете се намират за 40 000 такива случайни матрици и се провежда тест на квадрат за броя на редове 32, 31, 30 и <= 29.

Както винаги, тестът се повтаря и към получените p-стойности се прилага KS тест, за да се провери дали те са приблизително еднакви (Brown, 2021):

Monkey tests: Последователно определен брой битове се третират като като "думи". Броят се припокриващите думи в потока. Броят на "думите", които не се появяват, трябва да следва известното разпределение. Името на този тест се основава на теоремата за безкрайните маймуни.

Count the 1s: Преброява се 1 бит във всеки последователен или избран байт. Броят се преобразува в "букви" и се брои появата на "буква" от пет буквени думи;

Minimum distance test: На случаен принцип се поставят 8000 точки в квадрат 10000 × 10000, след което се намира минималното разстояние между двойките. Квадратът на това разстояние трябва да бъде разпределен експоненциално с определена средна стойност.

Random spheres test: На случаен принцип се изберат 4000 точки в куб с ръб 1000. Центрира се сфера върху всяка точка, чийто радиус е минималното разстояние до друга точка. Обемът на най-малката сфера трябва да бъде разпределен експоненциално с определена средна стойност.

The squeeze test: Умножава се 2^{31} с произволни число с плаваща запетая с (0,1), докато достигнете 1. Повторете това 100000 пъти. Броят на числата с плаваща запетая, необходими за достигане на 1, трябва да следва определено разпределение.

Overlapping sums test: Генерира се дълга последователност от произволни числа с плаваща запетая с (0,1). Добавят се редици от 100 последователни числа с плаваща запетая. Сумите трябва да бъдат нормално разпределени с характерни средни стойности и отклонения.

Runs test: Генерира се дълга последователност от произволни числа с плаваща запетая (0,1). Брои се по възходящи и низходящи трасета. Броят им трябва да следва определен ред.

The craps test: Играят се 200000 игри на зарове, като се броят печалбите и броят хвърляния на игра. Всеки брой трябва да следва определен ред.

Следва представяне на резултатите от теста с Dieharder на редицата случайни числа, като се включват 114 теста и прилагането им към различни криптографски операции. Качеството на стойностите, приложени в криптографията от хостинг сървъра, е под нивата на надеждна криптографска и киберсигурност на FIPS и теста за случайни числа FIPS-140 и други световни лаборатории **(Приложение 1).**

Разбор на данните от теста за симулация на случайни числа:

• Само 25 теста са преминали успешно;

• Неуспешни, които имат компрометирана /предсказуема/ стойност и следователно откриваема криптография са 76;

• Уязвими, където криптографията може да бъде разкрита с относително добър компютърен хардуер са 13;

От представените резултати може да се направи заключение, че заради слабостите в случайните числа и при установеното нарушаване на криптографската защита, рискът за успех при кибер атаки за компроментиране на криптографията е висок. Ако пробивът в криптографията сполучи, може да се извършат още множество опасни хакерски атаки, като някой по-известните такива са подмяна на съдържание за заблуждава на потребителя, фалшиви новини, изтичане на лични данни, достъп до други системи използвани от потребителя, посегателство върху сървърни ресурси и др.

Но проблемът е, че при текущото технологично състояние, предоставените публични хостинг услуги винаги ще имат установените фундаментални недостатъци. Предимство е, че са изключително достъпни, лесни за конфигуриране и изгодни. Ползите от тях не бива да се пренебрегват и те са от ключово значение за множество потребители. Но от представените резултати може да се заключи, че услуги с такова ниво на кибер сигурност, е нежелателно да се използват за системи с критична функционалност или работещи с лични данни. За тях е по-добре да се осигури хостинг на собствен сървър или нает виртуален частен сървър (VPS). Където броят едновременно работещи системи няма да надвиши възможностите на споделените системните ресурси използвани за обезпечаване на криптографията.

Друго възможно решение е да се опита по софтуерен път да се повиши качеството на ентропия на хостинг сървъра. Но при системи от този тип, често събирането на ентропия с помощни софтуерни средства е по-трудно от да кажем други тип работни станции. Защото те са лишени от периферия и други активности, които помагат и често активността на хардуера не е достатъчна за по-бързо обогатяване на ентропията на случайните числа. За това се препоръчва да се подпомагат от някаква допълнителна хардуерна активност. Представяме един пример за такава настройка на

виртуална машина с актуалната към момента на писане на този текст Linux Debian 10 операционна система:

Поради виртуализацията, буферът за ентропия стойността е изключително недостатъчна за нуждите на една система:

```
$ cat /proc/sys/kernel/random/entropy_avail
108
```

Още една проверка с командата `pv`:

```
# pv /dev/random > /dev/null
  40 B 0:00:15 [   0 B/s] [                    <=>                 ]
  52 B 0:00:23 [   0 B/s] [                           <=>          ]
  58 B 0:00:25 [5.81 B/s] [                              <=>       ]
  64 B 0:00:30 [6.05 B/s] [                                   <=>  ]
^C
```

Инсталиране на софтуер, предназначен да обогати по софтуерен път ентропията:

```
# apt install haveged
# systemctl start rngd
# update-rc.d haveged defaults
# rngd -r /dev/urandom
# systemctl start haveged
```

Ето и експерименталните резултати, след направените промени:

```
$ cat /proc/sys/kernel/random/entropy_avail
3226

# pv /dev/random > /dev/null
7.12MiB 0:00:05 [1.43MiB/s] [                <=>                   ]
15.7MiB 0:00:11 [1.44MiB/s] [                        <=>           ]
27.2MiB 0:00:19 [1.46MiB/s] [                         <=>          ]
  43MiB 0:00:30 [1.47MiB/s] [                                  <=> ]
^C
```

От резултатите е видно, че буферът е натрупал достатъчен капацитет от ентропия. Резултатите с командата pv са не по-малко впечатляващи. Може да се види, преди стартиране на haveged скоростта е 2,1 бита в секунда (B/s), докато след стартиране на haveged и добавяне на трептене на процесора към пула с Ентропия се получават ~ 1,5 MB / сек. Има и една малко по-нестандартна мярка, която може да се приложи. Като се добави допълнително софтуерно решение, което създава хардуерна активност, която обогатява събирането на ентропия. За конкретния случай е използван скрипт, който е подходящ за сървърни системи, които нямат периферия или тя не се използва:

```
#!/bin/sh
## list of sites using round-robin DNS
ROUND_ROBINS="www.yahoo.com google.com twitter.com outlook.com"
## Entropy stard and end value limits
STOP_LIMIT="3800"
START_LIMIT="3000"


until [ "$(cat /proc/sys/kernel/random/entropy_avail)" -gt
"$STOP_LIMIT" ]
    do while [ "$(cat /proc/sys/kernel/random/entropy_avail)" -lt
"$START_LIMIT" ]
        do for thing in "/tmp/loyeyoung" "/tmp/sueellen"
"/tmp/rootdev" "/tmp/files"
            do echo $thing ====================================
                touch /tmp/toss
                for robins in $ROUND_ROBINS
                    do nslookup "$robins" 8.8.8.8 > /tmp/toss
                        nslookup “$robins” 9.9.9.9 >> /tmp/toss
                        nslookup “$robins” 192.168.2.3 >> /tmp/toss
                        nslookup “$robins” >> /tmp/toss
                        cat /tmp/toss
                        mkdir $thing -p
                        cp /tmp/toss $thing/toss
                        cat $thing/toss
                        rm -f /tmp/toss
```

```
                    rm -f $thing/toss
                done
            done
        done
    done
```

Този програмен скрипт е съвсем базов и би могъл да бъде надграждан и съставян и на други програмни или скриптови езици. Въпреки семплия вид, успява да даде очакваните резултати за нуждите на текущото изследване. Скоростта на натрупване на ентропия се подобри. Което допринася въпросната система да понася по-големи натоварвания върху генерирането RNG стойности. Начинът на действие е както е зададен в момента е, че изпълнението на допълнителните операциите в памет, процесор, диск и мрежа, ще се активират при достигане на стойност в буфера за ентропия под 3000. Процесът ще спре при достигане на стойност по-висока от 3800:

```
$ cat /proc/sys/kernel/random/entropy_avail
3820
```

Макар и представеното решение да изпълнява успешно поставената задача, не би било достатъчно да попълни нуждата от случайни числа при натоварен сървър за споделен хостинг. Наистина до известна степен ще подобри допринесе, но не може да реши проблемите. Защото източниците консумиращи на случайни числа биха надхвърлили скоростта на набиране в буфера. Но такова решение би могло да се използва в комбинация и с хардуерни решения, което и компанията Intel предлага при своите процесори.

Наименованието на модула за подпомагане генерирането на случайни числа е Intel Secure Key, предишното му кодово име е Bull Mountain Technology. С това името Intel определя в процесорите си разширението за архитектура Intel64 и IA-32 RDRAND и свързаната с него хардуерна реализация на Digital Random Number Generator (DRNG). Освен всичко друго, DRNG, използвайки инструкцията RDRAND може да е изключително полезен при генериране на висококачествени ключове за криптографски протоколи. Следователно, за да може да се използва това решение, трябва да се направи проверка, дали използвания процесор предлага това разширение. В Linux с помощта на командата lscpu, може да се види информацията за процесорът на компютърната система:

```
$ lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
Address sizes:       39 bits physical, 48 bits virtual
CPU(s):              8
On-line CPU(s) list: 0-7
Thread(s) per core:  2
Core(s) per socket:  4
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               158
Model name:          Intel(R) Xeon(R) CPU E3-1505M v6 @ 3.00GHz
Stepping:            9
CPU MHz:             998.758
CPU max MHz:         4000.0000
CPU min MHz:         800.0000
BogoMIPS:            6000.00
Virtualization:      VT-x
L1d cache:           32K
L1i cache:           32K
L2 cache:            256K
L3 cache:            8192K
NUMA node0 CPU(s):   0-7
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep
mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht
tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon
pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni
pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16
xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer
aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb
invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority
ept vpid ept_ad fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms
invpcid rtm mpx rdseed adx smap clflushopt intel_pt xsaveopt xsavec
```

```
xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_window
hwp_epp md_clear flush_l1d
```

В последния ред от резултатът командата lscpu предоставя флаговете на поддържаните разширения на процесорът. Флагът обозначаващ поддръжката на Intel Secure Key от процесорът е логично с името RDRAND, който е маркиран с по-дебел шрифт и в резултатите по-горе. За по-кратко изписан резултат, който ще е подходящ за подпомагащи ентропията приложения, би могло да се използва и следната комбинация от команди:

```
$ cat /proc/cpuinfo | grep -i rdrand | echo $?
0
```

Като резултат 0 означава, че е наличен флаг RDRAND и процесорът може да бъде включен за подобряване на криптографските функции на системата по следния начин:

```
# apt install rng-tools-debian
# /etc/init.d/rng-tools-debian start
# /etc/init.d/rng-tools-debian status
* rng-tools-debian.service - LSB: rng-tools (Debian variant)
   Loaded: loaded (/etc/init.d/rng-tools-debian; generated)
   Active: active (running) since Fri 2020-11-28 17:30:54 EET; 3min
10s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 4 (limit: 4915)
   Memory: 1.3M
   CGroup: /system.slice/rng-tools-debian.service
           `-3597 /usr/sbin/rngd -r /dev/hwrng

$ cat /proc/sys/kernel/random/entropy_avail
4096
```

Резултатите показват, че скоростта на събиране на ентропия за нашият случай надхвърля скоростта на нейното консумиране. Но представената в това изследване система е актуална към момента на писане на този текст мобилна работна станция, която използвана за съвременен Intel Xeon процесор и софтуерните приложения

работещи върху нея не надхвърлят капацитета и на генератора на случайни числа и средствата за събиране на ентропия. Редно е всяка една сървърна система, да се извършва анализ на нивото на използване и натрупване на случайните числа. Ако въпреки всички представени тук начини за оптимизация на хардуер и софтуер отговорни за това и стойностите са недостатъчни. Трябва да се предвидят още машини за разпределянето на услугите и приложенията върху тях или добавяне на допълнителен хардуерен модул за генериране на случайни числа (TRNG).

3.4 Резултати в реална технологична инфраструктура

Предложения подход за подобряване на кибер сигурността в криптографията и генераторите на случайни числа при натоварени сървърни системи с публични услуги е приложен в технологичната инфраструктурата на института ИИКТ-БАН. Използваната хардуерна конфигурация е от среден клас, като е съобразена със сложността на изпълняваната задача. Сървърът е оборудван с един шест ядрен процесор Xeon(R) E-2236 от второ поколение и версия 6, 32GB RAM и два твърди диска в конфигурация с RAID1. Оперативния сървър с публичните услуги функционират върху Linux и всичките услуги са изцяло и от софтуер с отворен код. Функционират върху виртуална машина, като физическата машина е само виртуален хост. Което е еквивалентно със ситуацията с разглежданите масови услуги, които са в предмета на текущото изследване за кибер устойчивостта на криптографската защита. Сървърните услуги изпълнявани от виртуалната машина са:

- мейл сървър, към момента с 242 потребителски акаунта. Достъпен чрез SMTP, POP3, IMAP, като всички те са защитени с криптографски комуникационен протокол TLSv1.2 и TLSv1.3. Удостоверяват се със сървърен сертификат за установяване на TLS сесии с асиметричен алгоритъм от типа елиптична крива secp384r1. Свързването до услугата не може да се осъществи без криптиране на комуникацията;

- Уеб мейл, който позволява на всичките 242 потребителя да оперират с пощата си и през уеб браузър. Комуникацията е защитена чрез криптографският комуникационен протокол TLSv1.2 и TLSv1.3. Удостоверяват се със сървърен сертификат за установяване на TLS сесии с асиметричен алгоритъм от типа

елиптична крива secp384r1. Свързването до услугата не може да се осъществи без криптиране на комуникацията;

- Уеб портал на Институтът по информационни и комуникационни технологии към Българската академия на науките, което е основното уеб пространство на института. Съдържа информация за дейността, два научни журнала, както и структурна информация. Комуникацията е защитена чрез криптографски комуникационен протокол TLSv1.2 и TLSv1.3 и сървърен сертификат за установяване на сесии със асиметричен алгоритъм с елиптична крива secp384r1. Удостоверяват се със сървърен сертификат за установяване на TLS сесии с асиметричен алгоритъм от типа елиптична крива secp384r1. Не се позволява свързване до услугата по не криптиран канал;

- Услуга за отдалечена администрация SSH с най-високата степен на криптографска защита, предлагана от протокола към момента. Идентификацията на потребител по SSH е само чрез криптографски ключове, не се допускат пароли;

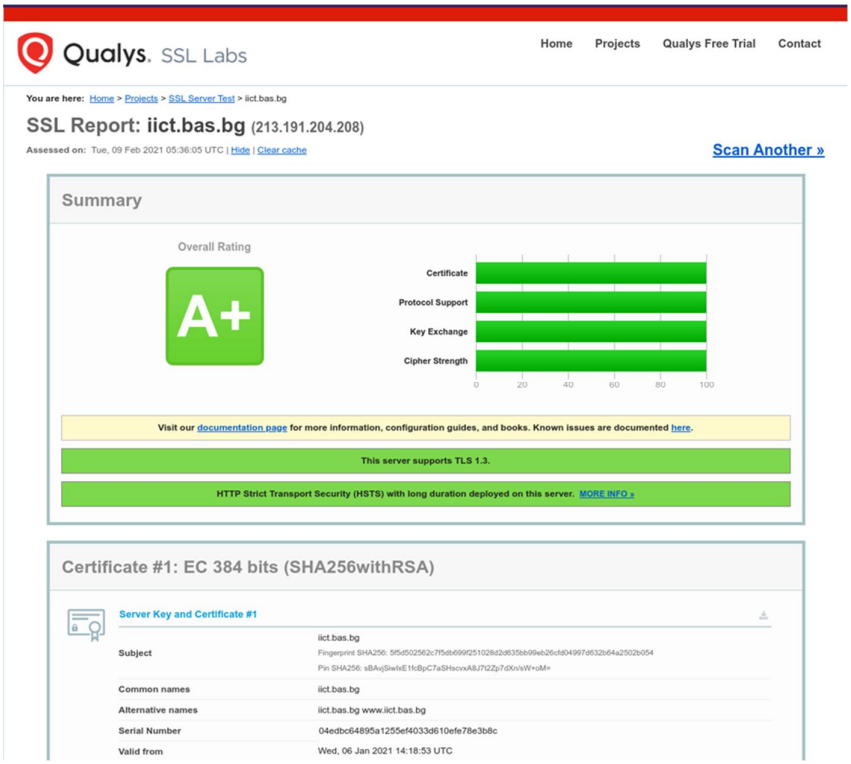
- Услуга за отдалечено управление на Уеб съдържанието FTP. Комуникацията е защитена чрез криптографски комуникационен протокол TLSv1.2 и TLSv1.3 и сървърен сертификат за установяване на сесии със асиметричен алгоритъм с елиптична крива secp384r1. Удостоверяват се със сървърен сертификат за установяване на TLS сесии с асиметричен алгоритъм от типа елиптична крива secp384r1. Не се позволява свързване до услугата по не криптиран канал;

За всички услуги приоритетен протокол за криптирана свързаност е най-новият и сигурен протокол TLSv1.3, но ако се окаже, че клиента не го поддържа се минава на протокол TLSv1.2. Последният е оставен само за съвместимост, като от него са премахнати всички криптографски алгоритми в които са открити уязвимости.

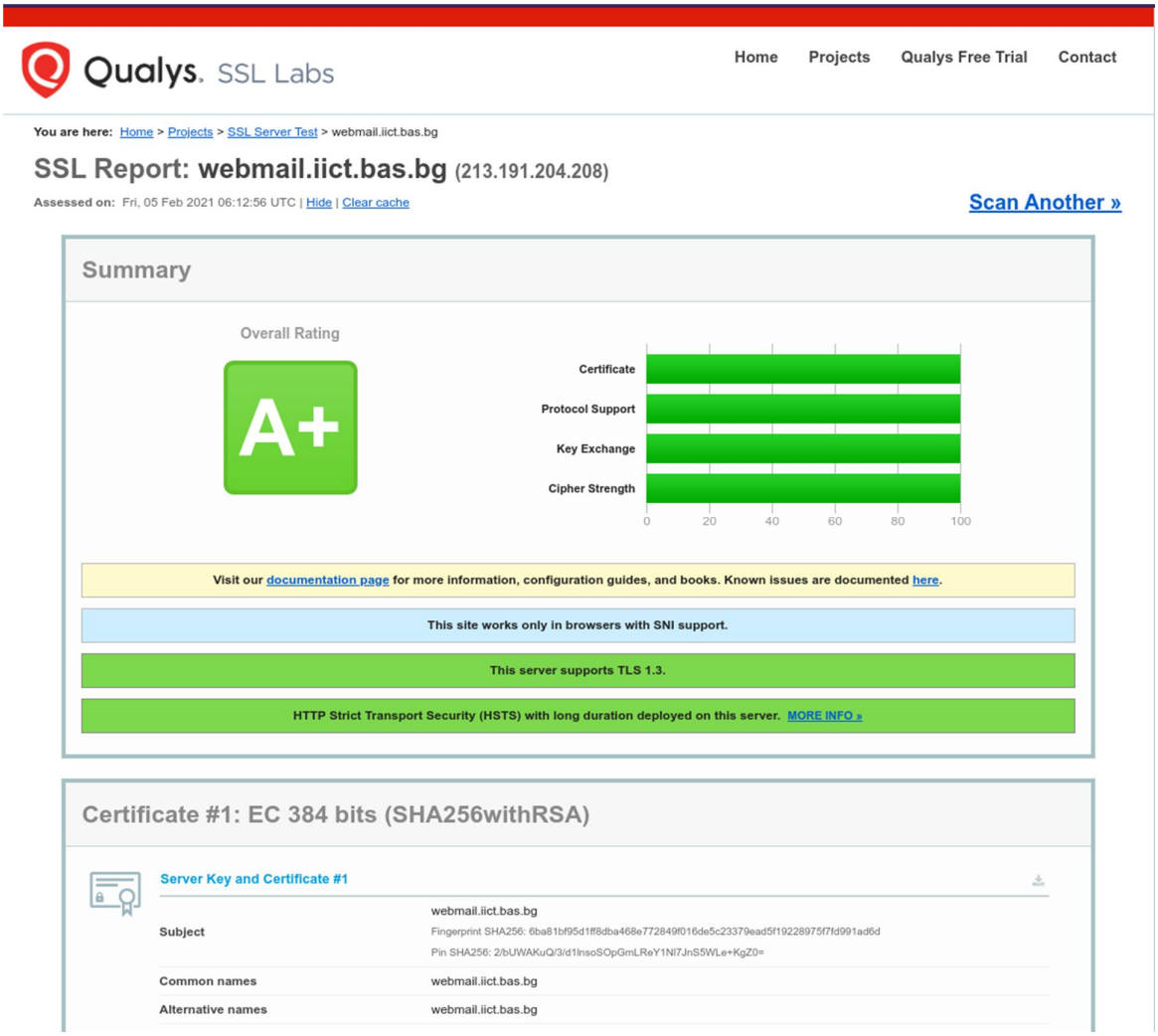
Освен това потребителите често използват едновременно повече от един начин за достъп до изброените услуги за един и същ пощенски акаунт. Защото пощата се получава в повечето случай на десктоп персонален компютър чрез мейл клиент. В същото време е настроена същата услуга и на смарт устройство – телефон/таблет. А може да се наложи и да се използва в някой случай Уеб браузър с Web-mail клиент. За това може да се каже, че натоварването на машината от страна на потребителски сесии

с необходимост от криптографска защита е значително. Което удовлетворява нуждите на среда за предмета на представеното изследването. Използването на криптографски алгоритми от по-висок клас, също изисква повече случайни числа. Защото приоритетно се използват по-дългите симетрични ключове от 256 бита, вместо както в повечето случай 128 бита и времето за тяхната работа е само в рамките на определена сесия. Макар броят на потребителите към момента на изваждане на тези резултати да е само 242, то по-качествената криптография изисква повече ентропия.

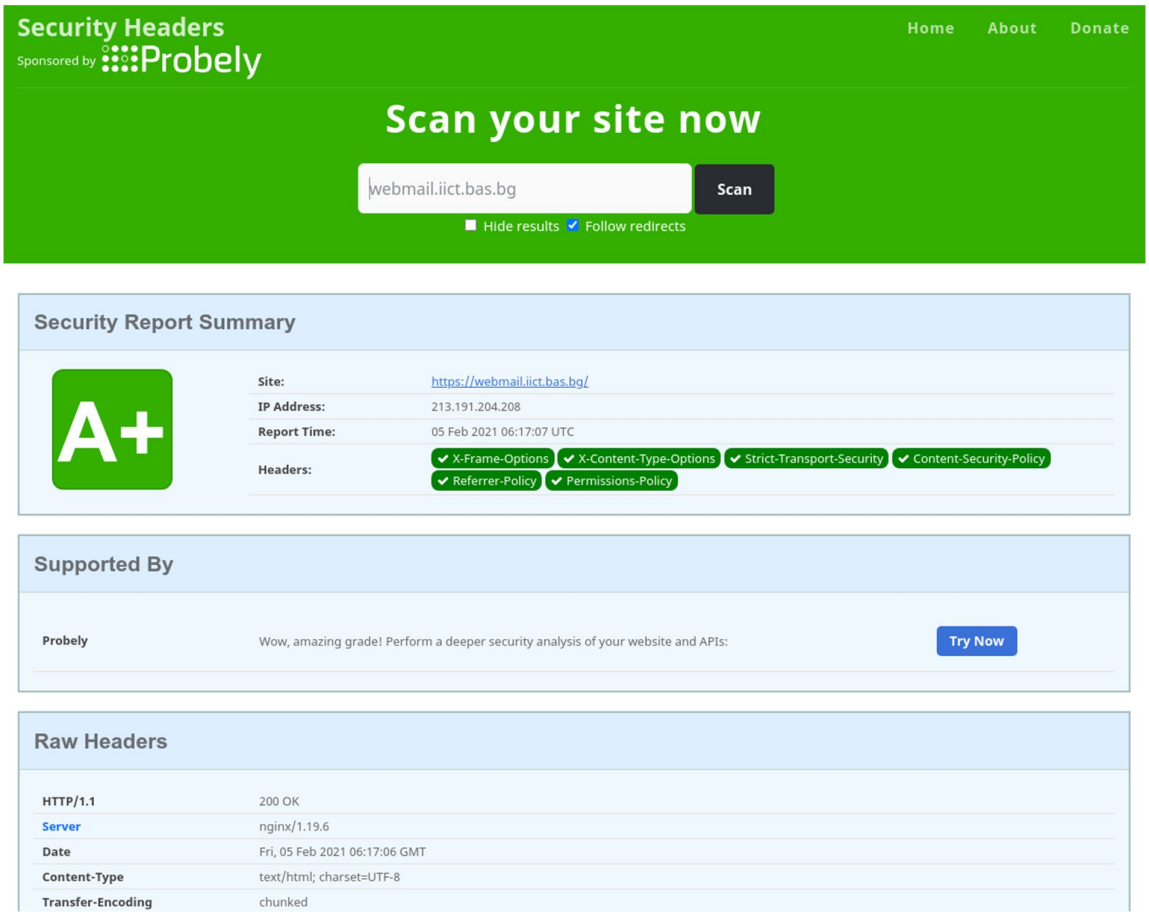
Като доказателство за качеството на кибер защитата с криптографски средства е приложен резултат от тест чрез скенер на SSL Labs, за нивото на криптиране при предлаганите TLS протоколи върху наличните услуги (фиг.3.7, фиг.3.8). Резултатите от този тест са извлечени на базата на актуалните изискванията за криптографска защита към момента, които са утвърдени от международните лаборатории по криптографска защита FIPS и NIST за САЩ, и Common criteria за Европа. От резултатите е видно, че протоколите и средствата за криптиране са от най-актуалните към сегашния момент. Оценката на всички тестове е най-високата възможна A+. Нивото на защита на HTTP протокола, който комуникира с браузъра, чрез TLS тунелът също е с максималното ниво на защита A+ (виж фиг.3.9)



Фиг. 3.7 Резултати от теста на криптографията от SSL Labs уеб портал https://iict.bas.bg



Фиг.3.8. Резултати от теста на криптографията от SSL Labs на уеб мейл https://webmail.iict.bas.bg



Фиг.3.9. Резултати от защитата на Уеб услугите на ниво HTTP протокол

Резултатите от анализа на протоколите за криптиране е още едно доказателство за необходимостта от достатъчно бърза и качествена ентропия. За осигуряването на която са използвани средства от предлаганите, като решение в изследването до момента. Изборът на процесор е в съгласие със съвременните изисквания и притежава силициев генератор за обогатяване на ентропията при случайните числа. Следното може да се види, след прилагането на командите представени до момента:

```
# lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
Address sizes:       39 bits physical, 48 bits virtual
CPU(s):              12
On-line CPU(s) list: 0-11
Thread(s) per core:  2
Core(s) per socket:  6
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               158
Model name:          Intel(R) Xeon(R) E-2236 CPU @ 3.40GHz
Stepping:            10
CPU MHz:             800.343
CPU max MHz:         4800.0000
CPU min MHz:         800.0000
BogoMIPS:            6816.00
Virtualization:      VT-x
L1d cache:           32K
L1i cache:           32K
```

```
L2 cache:              256K

L3 cache:              12288K

NUMA node0 CPU(s):     0-11

Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall
nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq pni pclmulqdq dtes64
monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1
sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand
lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single pti ssbd ibrs ibpb
stibp tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1
hle avx2 smep bmi2 erms invpcid rtm mpx rdseed adx smap clflushopt intel_pt
xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify
hwp_act_window hwp_epp md_clear flush_l1d

# cat /proc/cpuinfo | grep -i rdrand | echo $?

0
```

Извършени са тестове и на качеството на ентропия, чрез инструментите използвани в текущото изследване. От командния shell на сървъра са приложени два утвърдени, първият проверява качеството на ентропия по FIPS с rngtest, а вторият с инструмента за анализ dieharder. Преди изпълнението на двата теста е проверено нивото на ентропия натрупано в буфера, чрез съответната команда:

```
# cat /proc/sys/kernel/random/entropy_avail

3219
```

- Резултати от тест качеството на ентропия с инструмента **rngtest**:

```
# rngtest -c 1000 </dev/random

rngtest 5

Copyright (c) 2004 by Henrique de Moraes Holschuh

This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
rngtest: starting FIPS tests...

rngtest: bits received from input: 20000032

rngtest: FIPS 140-2 successes: 1000

rngtest: FIPS 140-2 failures: 0

rngtest: FIPS 140-2(2001-10-10) Monobit: 0

rngtest: FIPS 140-2(2001-10-10) Poker: 0

rngtest: FIPS 140-2(2001-10-10) Runs: 0

rngtest: FIPS 140-2(2001-10-10) Long run: 0

rngtest: FIPS 140-2(2001-10-10) Continuous run: 0

rngtest: input channel speed: (min=643.343; avg=12710.535;
max=16509.932)Kibits/s

rngtest: FIPS tests speed: (min=47.329; avg=224.928; max=244.532)Mibits/s

rngtest: Program run time: 1621642 microseconds
```

- Резултати от тест качеството на ентропия с инструмента **dieharder** (виж в Приложение.1 Dieharder entropy improvement results). При 114 теста върху случайните числа, с успешен резултат са всички и нито един не е с резултат провален или уязвимост.

Всички тези тестове са изпълнени по време на работа на сървъра в реалната репродукционна среда. Размера на буфера за ентропия, който е наличен в операционната система е с размер 4096 бита, както при всички останали Linux дистрибуции. Към момента на тестовете, нивото на натрупана ентропия е 3219 бита. Което показва, че в момента се използват случайните числа, иначе би стигнало близо до максималните стойности около 4000.

Въпреки добрите резултати е добре да се направи още едно изследване, според което ще стане ясно, дали потребителската активност и интензивното натоварване на криптографията ще доведат до изчерпване капацитета на случайните числа. За целта е съставен скрипт, който е добавен като задача в Cron, която на всеки 10 минути записва размерът на натрупаната в буфера ентропия. При период от време половин месец, ще

стане ясно дали има моменти, които водят до източване на буфера с ентропия по-бързо от колкото неговото зареждане от системата.

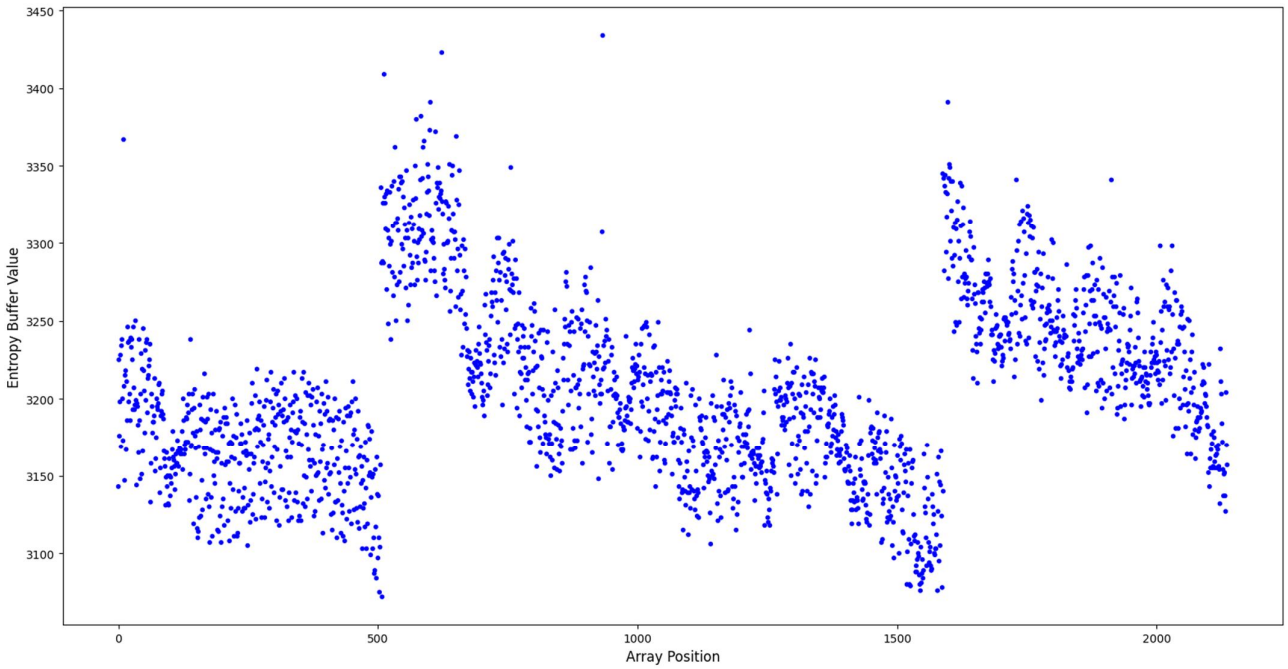
- Скрипт за извеждане стойността на буфера с ентропия и добавянето към файл:

```
#!/bin/bash

log_location="/var/log/"

log_file="entropy.log"

cd /root/ && cat /proc/sys/kernel/random/entropy_avail >>
$log_location$log_file
```

- Добавяне на задачата към cron за изпълнение на всеки 10 минути:

```
*/10 * * * *     /root/entropy.log.sh
```

Описаното действие доведе до създаване на времеви ред, с 2137 стойности. Тези стойности са изведени в графичен вид и са изобразени на фиг.3.10:



Фиг.3.10 Ниво на ентропия в различни моменти от времето

От графиката на фиг.3.10 личи, че системата е имала пикове на по-интензивна дейност, които са водели до силно използване на натрупаната в буфера ентропия. За това стойностите на моменти рязко падат. Приложеният в изследването подход върху системата обаче, успява да компенсира високата консумация на ентропия. Макар на графиката да се виждат широки амплитуди, то стойностите са в не много широк

диапазон, с ниво на ентропия между 3000 и малко под 3450. Липсата на стойности под 3000 показва, че системата се намира в много добро здраве по отношение на случайните числа. Дори е способна да поеме и по-интензивни натоварвания, защото е далеч от стойностите на изчерпване. Тестовете на ентропията показаха възможно най-добрите резултати. Като се вземат предвид всички тези резултати, е на лице доказателството за ефективността на предлагания подход. Използването на най-добрите към днешно време криптографски шифри, също не успяват да изчерпят критично средствата за ентропия на системата. Интензивните натоварвания от над 200 потребителя, също не успяват да изчерпят качеството на случайните числа в системата и да доведат до уязвимости в криптографията и кибер сигурността. Следователно може да се счита, че предлаганият подход може да бъде от полза и да подпомага различните Интернет системи и решения.

3.4 Изводи

Изследването на представените услуги и тяхното ниво на киберсигурност е от ключово значение по-сигурен и бърз преход към съвременната дигитална трансформация. Бързото прехвърляне на всички социални и икономически дейности към дигитални платформи доказва, че съществуващата технологична инфраструктура може да отговори на днешните предизвикателства за дигитална трансформация. Ползите от това в икономическо и екологично отношение са неоспорими. Но по отношение на киберсигурността, много от настоящите ИТ услуги все още изостават. Увеличаването на степента на успех на кибер престъпленията може да доведе до загуба на доверието в технологиите и възпрепятстването на тези процеси, което ще засегне и на научно-техническия прогрес. Също така ще повлияе на забавянето в развитието и на много други свързани области в икономиката, сигурността, технологиите и други. Има и нещо още по-важно, регистрират се все повече организирани кибер атаки от ново поколение. Така наречения кибер тероризъм, това са мощни терористични организации които действат в цифровото пространство и нанасят големи вреди на държави, организации и дейности. За това може да се счита, че този въпрос е още по-важен, защото вече е свързан и с националната сигурност и не може да се разглежда само, като част от битовата престъпност.

Прилагането на математически и статистически анализи с времеви редове за решаване на проблеми в кибер сигурността е ефективно. Предлаганите тук подходи, може да се комбинират и с други техники и методи за анализ на кибер сигурността, за да са по-комплексни и ефективни. Ще бъде добре дори и в бъдеще да се изграждат цели системи за одит и анализ на кибер сигурността, които да са оборудвани с множество такива методи за изследване и намиране на слабости и към тях да се съдържа и решения за премахването им. Във времето тези анализи и решения ще бъдат развивани и разширявани. Освен, че съществуващите системи ще бъдат значително по защитени, ще се утвърдят и добри практики в кибер сигурността, които ще се внедряват при новите разработки и технологични решения във всички технологични области.

В тази дисертационна работа е разработен метод за изследване на качество на RNG и PRNG в информационна система чрез прилагане на времеви редове. Методът позволява да се повиши качеството на ентропия при използването на криптография, осигуряваща различни Интернет услуги. Разработен е алгоритъмът за откриване на повтарящи се модели (patterns) от данни в генерираните от RNG времеви редове. Проведено е изследване на криптографски тестове и качеството на ентропия върху работещи в реални условия натоварени сървърни системи с публични Интернет услуги.

Съдържанието на тази глава е отразено в публикациите:

1 **Blagoev I.,** Method for more reliable users' authentication in internet, Сборник доклади от межд. конференция, НВУ "Васил Левски", 14-15 юни 2018, Том 9, стр. 167-176. E-ISBN-13: 978-619-7246-20-9

2 **Ivan Blagoev**, Method for Evaluating the Vulnerability of Random Number Generators for Cryptographic Protection in Information Systems, международна конференция HIGH PERFORMANCE COMPUTING - BULGARIA 2019, Borovets, Bulgaria, 2-6 September 2019, Springer, Studies in Computational Intelligence, **SJR 0.183,** ISSN18609503, 1860949X

3 **Ivan Blagoev**, Application of Time Series Techniques for Random Number Generator Analysis, Proceedings of XXII Int. Conference DCCN 2019, September 23-27, 2019, Moscow, Russia, pp.437-446. ISBN 978-5-209-09683-2, 2019 (РИНЦ).

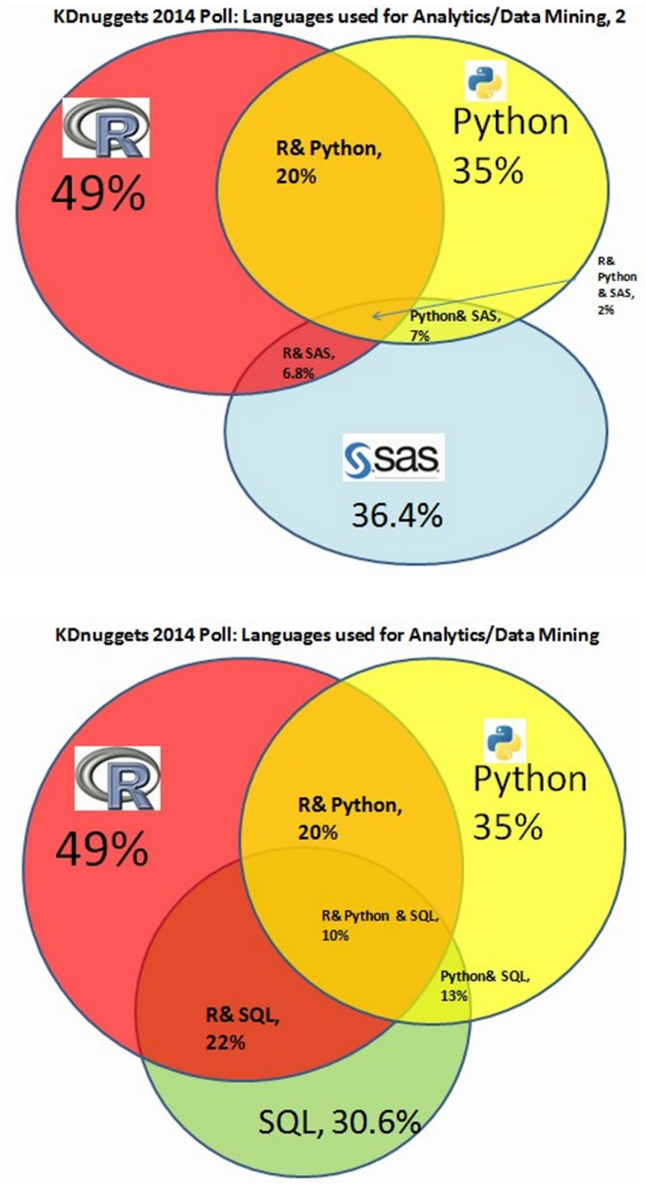
4 **Blagoev I.,** Neglected Cybersecurity Risks in the Public Internet Hosting Service Providers. *Information&Security International Journal* - ISIJ, 47, no. 1, pp. 62-76 (2020) Print ISSN 0861-5160 (previously 1311-1493) Online ISSN 1314-2119 http://dx.doi.org/10.11610/isij.4704

Глава4. Софтуерни подходи при работа с големи масиви от данни и ограничени компютърни ресурси с език за програмиране R

4.1 Програмният език R

Програмният език R е продукт, разполагащ с мощни инструменти за статистически изчисления и анализи. R едновременно е програмен език и софтуерна среда (Borcard, 2011), (The R, 2017), (Venables, 2016). Компилира се и работи на различни операционни системи, като UNIX платформи, Linux, Windows и MacOS. Езикът е създаден е през 1996 г. от Рос Иака и Робърт Джентълмен. Названието му („R”) произлиза от началните букви на малките имена на неговите създатели и, както се разбира от неговото пълно име – „Програмен език и развойна среда за статистическа и математическа обработка на данни”, той е мощен инструмент използван от хиляди учени, преподаватели и студенти по света от най-различни дисциплини и направления. През последните години изучаването на езика се превръща в стандарт в катедрите по статистика и математика на повечето от университетите в Западна Европа и САЩ (Фиг. 4.1). Също сред поддържащите и използващите езика са компании като: Google, Astra, Merck, AT&T Labs, Baxter Healthcare Corporation и много други. Изследователите в Google признават: „R е толкова важен за компанията, че е трудно да си представим какво би било ако него го нямаше. Той позволява на учените да провеждат сложни анализи без да е необходимо те да притежават задълбочени познания в областта статистиката, математиката, както и на компютрите и компютърните системи”.

Причината, поради която езикът R е придобил популярността си, е в неговия интерактивен език, който улеснява изследването, изясняването и представянето на данните (Patil, 2016). Пълният работен процес за изследване на данни, във вид на набор от стъпки за използване на конкретни му софтуерни пакети, е представен в (Wickham, 2016), осигурявайки общ ресурс за R. Прегледът на методи за анализ на данни с R е направен в (Long, 2015).



Фиг.4.1. Четири основни езика за Анализи, Data Mining, Data Science (Four main languages for Analytics, Data Mining, Data Science, 2014, http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html)

Езикът R (Фиг. 4.2) предлага широко разнообразие от статистически техники като, например линейно и нелинейно моделиране (Douc, 2014), класически статистически тестове, анализ на времеви редове, класификация, групиране и др., както и графични техники и е изключително разширяем (Long, 2015).

Средата за разработки е логически добре подредена и може да се определи, като лаборатория за статистическо изследване. Функциите, налични за потребителя, се намират в библиотеки, които са разположени в директорията „R_HOME/library” („R_HOME” е така наречената директория - корен на средата, където „R” се инсталира). Там се съдържат групи („packages“) от функции, които също са структурирани в съответните директории. Основната група от функции в „R” се нарича „base” (база) и съдържа в себе си функциите за четене, обработка и представяне на информацията. Езикът включва възможности за работа с различни типове данни - както числови, така и низови и логически (булеви). Обектните структури са като: вектори, фактори, списъци, матрици и дейта-фреймове. Възможностите на R в това отношение понастоящем се използват широко при изграждането на математически симулации и модели на различни биологични и природни системи.

Фиг. 4.2. R Лого © 2016 The R Foundation

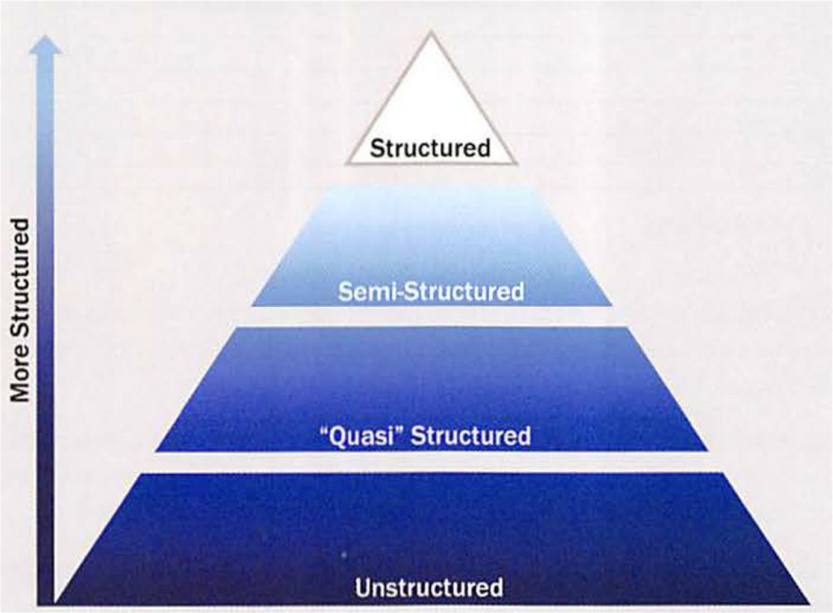
Запознаването с езика за програмиране R (Toomey, 2014) позволява по достъпен начин да се овладеят основите на логиката, изграждаща компютърната програма, и основите на описание на данните, включително фактори, списъци и дейта фреймове (data frames). Във всяка една област на науката, това би допринесло за по-прецизни анализи, изследвания и би довело до по-добри резултати. Това позволява и на непрофесионални програмисти, които да не познават добре функционирането на компютърната система и изпълнението на програмния код, да могат да извършат сложен анализ на данни с множество вътрешни сечения. R също така съдържа в себе си много мощни инструменти за графично визуализиране на данни, чрез изчисления и анализи върху тях.

Въпреки многото предимства на R, богатството на неговите статистически модели и инструменти за обработка на данни, както и мощните способности за визуализация, изникват проблеми при работа с големи обеми от данни, чиито характеристики са показани на фиг. 4.3. Нарастването на количеството данни и начинът им на обработване доведоха до сериозни ограничения при работа с R. Ограниченията произлизат от това, че той е проектиран да оперира в режим на изчисления само в единичен процес (на единично ядро на процесор) и при данните, заредени наведнъж в оперативната памет.

Технология	Хранилище на големите данни	Скорост	Мащабируемост	Изчисляване в паметта	Опит на стандартен R потребител
Bigmemory Biglm parallel	Не	Много бавна	Ниска	Не	Да
RDBMS със статистическа поддръжка	патентовано	Висока, ако е присъща	Средна	Варира	Не
R on MapReduce	HDFS	бавна	висока	Не	Не

Табл. 4.1 Сравнение на различни технологии.

Сравнение на различни технологии (Табл. 4.1) показва, че неговите предимства са високата изчислителна ефективност, но недостатъкът е ограниченото количество данни в паметта и времето им за обработка. Бързото развитие на комуникациите и технологиите от последните години доведоха до едно ново предизвикателство – големите данни. Генерирани от множество системи, тези данни са необятно поле за усъвършенстване на технологии и развитие на науката. Следователно за R се явяват все повече задачи, които не могат да бъдат решавани в рамките на определена компютърна система, тъй като R се срива при работа с големите данни (Mahmud, 2020).



Фиг. 4.3. Характеристики на големите данни (Long, 2015).

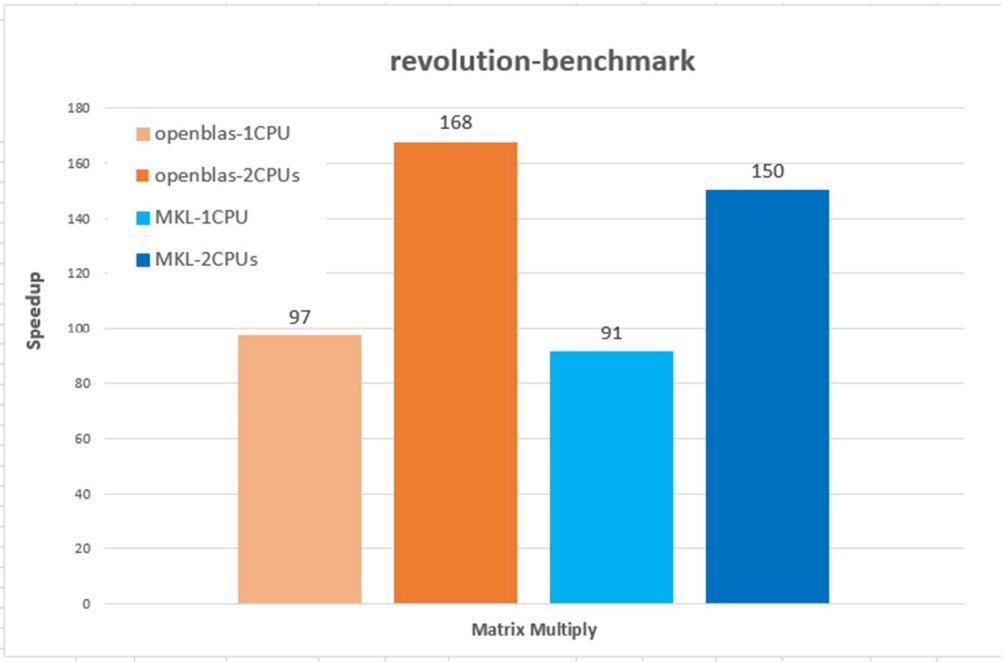
Съвременните компютърни системи притежават повече от едно изчислително ядро в микро-процесора си и не малко оперативна памет, която обаче е недостатъчна за възможностите на големите данни.

4.2 Преодоляване на проблеми на работа с големите данни чрез използване на микропроцесор с много ядра

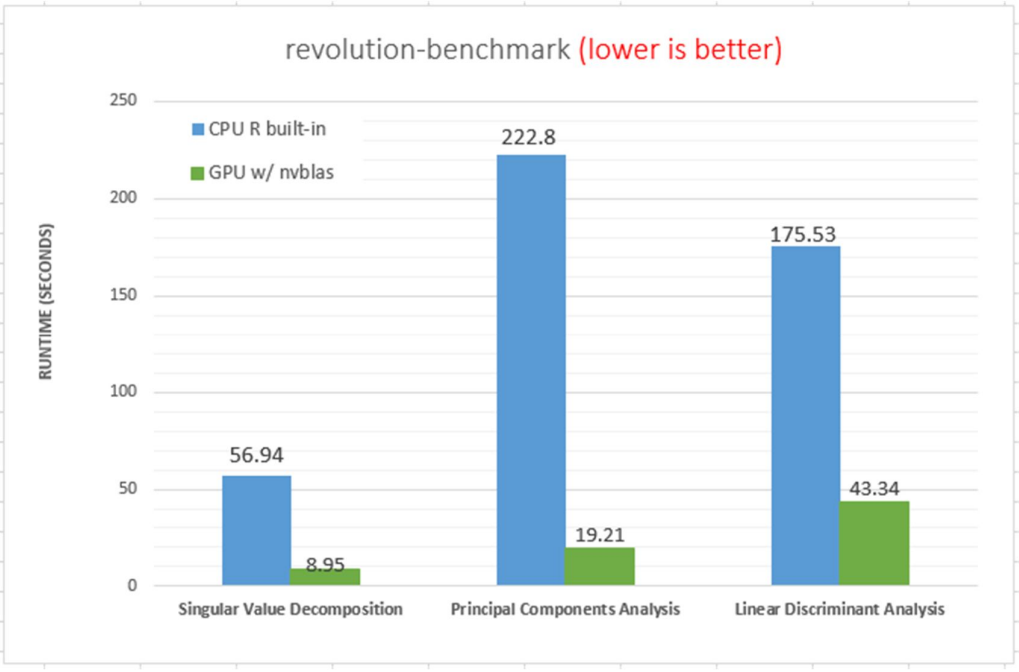
Паралелното програмно изчисляване на повече от едно ядро на процесор е възможно чрез прекомпилиране и добавяне на някои програмни компоненти в R. Това е възможно, поради факта, че R е система с отворен код и това е едно от предимствата, което носи тази концепция. Разнообразието от решения на този проблем зависи от разнообразието при производителите на компютърни компоненти и различните проекти с отворен код. За това не е възможно да изброим всички възможности, а само някои по-разпространени от тях:

- Intel® Math Kernel Library (Intel® MKL);
- NVIDIA cuBLAS;
- OpenBLAS.

В тези програмни библиотеки са разработени различни векторни функции за паралелизация на процесите, които увеличават значително производителността, като намаляват времето за обработка на данните. Използвани са също така индустриални стандарти за разработката на библиотеките и интеграцията с интерпретатора на R, което позволява да не се променя кода, написан на R. Разликата в повишение на производителността при една и съща компютърна система, може да се види на Фиг. 4.4 и Фиг. 4.5:



Фиг. 4.4. Сравнение на производителността при умножение на матрици.



Фиг. 4.5. Сравнение по бързодействие.

4.3 Методи за оптимизиране на обеми от данни

При наличие на голям файл с данни, може да се редуцира зареждането им в паметта, като към момента на зареждане се изключат редовете с некоректно съдържание. При работата с големите данни наличието на редове, които са повредени или с некоректни стойности е често срещано. Така, анализирайки входящия поток от данни, ще се отделят тези, които не са необходими за изпълнение на конкретната задача.

- *Изключване на редовете с некоректно съдържание.*

В този случай колони без числови стойности (NA) ще ги приемем за такива. Нека файлът, който зареждаме е с име „envdata_raw.csv“ и има следното примерно съдържание:

```
"X","Y","Z"
1,2,3
1,NA,4
4,6,7
NA,NA,NA
4,8,NA
```

Тези редове може да се премахнат при зареждането на файла по следния начин:

```
>env_data=na.omit(read.csv("envdata_raw.csv", na.strings=c("",
"NA")))
> write.csv(env_data, file="envdata.csv")
> print(env_data)
     X     Y     Z
1    1     2     3
3    4     6     7
```

В резултата личи, че редовете 2, 4 и 5 не са заредени в паметта, променливата `env_data` съдържа само редовете с валидните за този случай числови данни. На втори ред е използвана команда за записване на файл с обработеното съдържание с име

„envdata.csv“. При следваща необходимост да се използват тези данни, може да заредим в R новият файл и така ще избегнем използването на компютърни ресурси за генерирането на същото съдържание.

Преди да се заредят данни от файл, може да се направи предварителен анализ на обема от информация, като брой редове, които ще бъдат предмет на обработка:

Изброяване на валидни редове в даден файл с данни, без да се зарежда цялото съдържание на файла:

```
>print(length(count.fields("envdata_raw.csv"))-1)
[1] 5
>print(length(count.fields("envdata.csv"))-1)
[1] 2
```

Изваждаме от резултата на length цифрата 1, за да избегнем броенето и на имената на колоните, като ред с данни.

В някои статистически изследвания не е необходимо да се зареждат всичките данни, а само определени времеви рамки, за да се направи приблизителен статистически анализ в отрязък от време. В такъв случай, може да използваме позиционирано прочитане и обработка на определен отрязък от данните, разположени във файла с големите данни:

Зареждане на определен брой редове от началото на даден файл:

```
> env_data=read.csv("envdata_raw.csv", na.strings=c("","NA"),
nrows=2)
> print(env_data)
     "X"   "Y"   "Z"
1    1     2     3
2    1     NA    4
```

В командата за четене е добавен параметър nrows=2, с което се казва на командата да прочете само първите два реда от избрания файл. Така на обекта „env_data“ присвояваме данни, само колкото сме определили и са необходими за определения случай.

Зареждане на произволен брой редове от края на даден файл:

```
>env_data=tail(read.csv("envdata_raw.csv",2)
> print(env_data)
      X     Y     Z
4     NA    NA    NA
5     4     8     NA
```

Чрез командата „tail“ и втори параметър със стойност 2, се оказва, че в обекта „env_data“ ще бъдат заредени само последните два реда от съдържанието на посочения файл.

Заличаване на обекти от паметта, които вече не използваме:

```
> x=tail(read.csv("envdata_raw.csv"),2)
>env_data=na.omit(read.csv("envdata_raw.csv",
na.strings=c("","NA")))
> ls(all.names = TRUE)
[1]   „env_data“ „x“
>remove(x)
>ls(all.names = TRUE)
[1]   „env_data“
>remove(env_data)
>ls(all.names = TRUE)
character(0)
```

При работа с R, може да се случи да заредим в паметта повече от един обект, като например масив с данни, таблици, матрици и др. В един даден момент това би довело до невъзможното зареждане на нови данни или генериране на нови резултати от изчисления. За това командата `remove()` веднага заличава ненужния обект от паметта. Възможно е да сме забравили в паметта на компютърната система създадени от нас обекти, които не използваме вече и са все още „живи“, за това в примера е добавена и командата `ls(all.names = TRUE)`. Така може да видим всички обекти, създадени от скриптове и команди и да премахнем ненужните от тях.

4.4 Изводи

Приносът на автора е, че чрез този материал се подпомага решаването на проблеми при работа с големи масиви от данни и ограничени компютърни ресурси със

средствата на език за програмиране R. В тази дисертация са разработени софтуерни техники за оптимизиране на компютърната памет при работата с големи данни.

В заключение може да се каже, че с представените до тук примери не може да се изчерпа темата за оптимизираното зареждане на данни при работа с език за програмиране R. Работа с реални данни винаги е предизвикателство (Baumer, 2017). Но представените до тук техники са между добрите практики и са често използвани, те биха могли да се комбинират и с други подходи за решаването на проблеми в тази област. Повдигането на този въпрос, също насочва мисленето на R потребителите към този проблем, това провокира желанието за търсене на подходи водещи до създаването на по-производителни решения и по-високо качество на изпълнените задачи.

Съдържанието на тази глава е отразено в публикациите:

1 **Ivan Blagoev**, Методи за оптимизирано използване на компютърна памет при зареждане на данни със средствата на език за програмиране R (Methods for Optimized Use of Computer Memory during Data Loads with R Programming Languages), Int. Conference Automatics and Informatics'2017, 4-6 October 2017, Sofia, Bulgaria, ISSN:1313-1850, pp.213-215.

2 **Blagoev, I.,** Using R Programming Language for Processing of Large Data Sets, Proc. Int. Conf. Big Data, Knowledge and Control Systems Engineering – BdKCSE'2018, 21-22 November 2018, Sofia, Bulgaria ISSN 2367-6450, pp. 91-98.

Заключение - резюме на получените резултати

В дисертационния труд подробно са изследвани методи и средства за използване на времеви редове при решаване на различни задачи, възникващи в съвременните приложения на информационни технологии и системи. Изследването се основава върху метода (Creswell, 2009), като използва смесени методи за проучване, комбинирайки и свързвайки и двата метода - качествен и количествен. При качествен метод фокусът е върху индивидуалното значение и важността на представянето на сложността на ситуация. А количественото изследване е приложено за тестване на обективни явления чрез проучване на връзките между променливите.

Предложен е метод озаглавен MA Volatility Indicator за подобряване прецизността в осцилатор (Моментум). MA Volatility Indicator работи в комбинацията от два инструмента EMA или SMA и предлага нова методика за интерпретиране на резултатите, което допринася за откриване на нива за свръх покупка и свръх продажба при пазарната тенденция. Всички използвани в изследването инструменти EMA, SMA и Моментум, както и MA Volatility Indicator използват времеви редове.

Разгледана е приложимостта на апарата на невронните мрежи за прогнозиране на времеви редове във финансовата област. Показано е, че с нов модел на представяне на входните данни, характерни за финансови показатели, се получава по-висока степен на самоадаптация при обучение на невронната мрежа. Проведените експерименти потвърждават сложността на финансовите процеси и наличието на високочестотен шум в данните.

Разработен е метод за изследване на качество на RNG и PRNG в информационна система чрез прилагане на времеви редове за да се повиши качеството на ентропия на при използването на криптография осигуряваща различни Интернет услуги. По този начин се допринася за по-добрата киберсигурност на ИТ инфраструктура за цифрови ресурси и защитата на данни. В дисертационната работа темата за криптографията получи специално внимание, поради нейното критичното значение. При пропускането и само на един риск в киберсигурността е възможно да бъдат компрометирани всички ИТ услуги.

Практическите резултати от реалния експеримент показаха, че е намерено златното съотношение между масови услуги и действителните изисквания за киберсигурност.

С оглед на работата извършена в този дисертационен труд и резултатите, получени в хода на изследванията и изложени по-горе, могат да бъдат формулирани следните **научно-приложни приноси**:

1. Разработен е метод озаглавен MA Volatility Indicator чрез комбиниране на индикатори за анализ и предсказване на ценови движения с нови подходи при използване на времеви редове от финансовите данни;

2. Разработен е алгоритъм за обучение на невронната мрежа при прогнозиране на финансови времеви редове чрез увеличаване на размера на входа на невронна мрежа и създаване на самонадграждащи се трислойни MLP.

3. Разработен е метод за повишаване на криптографската защита в информационните системи на базата на изследвания на качеството на генераторите на произволни числа чрез прилагане на методи за анализ на времеви редове.

4. Проведени са експериментални изследвания за верификация на предложените методи за решаване на проблемите с киберсигурността в публични широко разпространени хостинг услуги. Получените резултати потвърждават валидността на предложения метод за повишаване на киберсигурността.

5. Разработени са програмни методи за ефективна работа с големи данни във времеви редове със средства на езика R.

6. Разработените методи за повишаване на криптографска защита са имплементирани в технологичната инфраструктурата на ИИКТ-БАН. Проведено е изследване на криптографски тестове и качеството на ентропия върху работещи в реални условия натоварени сървърни системи с публични Интернет услуги.

Насоки за бъдещи изследвания

Насоките за бъдещи изследвания по тематиката на дисертацията включват:

- Имплементиране на методът MA Volatility Indicator и прилагането му в комбинация и с други методи за анализ и прогнозиране на пазарни ценови тенденции;
- Прилагане на методът MA Volatility Indicator към автоматизирани системи за анализ на пазарни тенденции и извличане на сигнали за взимане на решения;
- Провеждане на още изследвания в областта на обучаващи алгоритми и системи с невронни мрежи за анализ и прогнозиране на времеви редове;
- Развиването на нови методи за увеличаване на криптографската защита в информационните системи;
- Изследване на комбинация на разработен метод с други методи и системи за анализ на RNG в криптография и други технологични области, което да съдейства за създаване и усъвършенстване на RNG, както и за по-точно определяне на спектъра от задачи, които генераторът може да изпълнява добре;
- Намиране на още подходи за зареждане и филтриране на големите данни с цел по-ефективната им обработка.

Публикации по темата на дисертационния труд

1 **Иван Благоев**, Николай Докев, Комбиниране на Моментум с Един Метод за Прогнозиране на Пазарни Ценови Движения За По-Точни Резултати (Combination of Momentum with One Method for Forecasting of Market Trends to Improve the Results), Международна научна конференция "УНИТЕХ'17" – Габрово, 2017 Selected papers, ISSN 2603-378X, pp. II-265-II-270

2 **Ivan Blagoev**, Методи за оптимизирано използване на компютърна памет при зареждане на данни със средствата на език за програмиране R (Methods for Optimized Use of Computer Memory during Data Loads with R Programming Languages), International Conference "Automatics and Informatics'2017", 4-6 October 2017, Sofia, Bulgaria, ISSN:1313-1850, pp.213-215.

3 **Blagoev I.,** Improving the Momentum Oscillator Accuracy by a Method for Forecasting of Market Price Movements, Сборник доклади от международна конференция, НВУ "Васил Левски", 14-15 юни 2018, Том 9, стр. 177-185. (ceeol.com)

4 **Blagoev I.,** Method for more reliable users' authentication in internet, Сборник доклади от международна конференция, НВУ "Васил Левски", 14-15 юни 2018, Том 9, стр. 167-176.

5 **Blagoev, I.,** Using R Programming Language for Processing of Large Data Sets, Proc. Int. Conf. Big Data, Knowledge and Control Systems Engineering – BdKCSE'2018, 21-22 November 2018, Sofia, Bulgaria ISSN 2367-6450, pp. 91-98.

6 **Ivan Blagoev**, Application of Time Series Techniques for Random Number Generator Analysis, Proceedings of XXII Int. Conference DCCN 2019, September 23-27, 2019, Moscow, Russia, pp.437-446. ISBN 978-5-209-09683-2, 2019 (РИНЦ).

7 **Blagoev I.,** Neglected Cybersecurity Risks in the Public Internet Hosting Service Providers. Information&Security International Journal - ISIJ, 47, no. 1, pp. 62-76 (2020)

8 Balabanov T.D., **Blagoev I.I.,** Dineva K.I. (2018) Self Rising Tri Layers MLP for Time Series Forecasting. In: Vishnevskiy V., Kozyrev D. (eds) Distributed Computer and Communication Networks. DCCN 2018. Communications in Computer and Information Science, vol 919. Springer, Cham. https://doi.org/10.1007/978-3-319-99447-5_50, pp. 577-584, **SJR:0.188**

9 **Blagoev I.** (2020) Method for Evaluating the Vulnerability of Random Number Generators for Cryptographic Protection in Information Systems. In: Dimov I., Fidanova

S. (eds) Advances in High Performance Computing. HPC 2019. Studies in Computational Intelligence, vol 902. Springer, Cham. https://doi.org/10.1007/978-3-030-55347-0_33 **SJR:0.215**

Забелязани цитирания

I Blagoev, I., 2018. Using R Programming Language for Processing of Large Data Sets, Proc. Int. Conf. Big Data, Knowledge and Control Systems Engineering – BdKCSE'2018, pp. 91-98.

Цитира се в:

1 Dineva, K., Atanasova, T.: Regression Analysis on Data Received from Modular IoT System. ESM'2019, EUROSIS-ETI, ISBN: 978-9492859-09-9, EAN: 9789492859099, pp.114-118, 2019

2 Ivaylo Blagoev, G. Vassileva and V. Monov, "Methodology for content preparation of online courses," 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-4, doi: 10.1109/ICAI50593.2020.9311364.

II Blagoev I., Neglected Cybersecurity Risks in the Public Internet Hosting Service Providers. Information&Security International Journal - ISIJ, 47, no. 1, pp. 62-76 (2020)

Цитира се в:

3 M Terzieva, D Karastoyanov, ICT for Innovation in Advanced Banking, PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2020 • Vol. 73, pp. 47-54 p-ISSN: 2738-7356; e-ISSN: 2738-7364, doi: 10.7546/PECR.73.20.05

III Blagoev I., Method for more reliable users' authentication in internet, Сборник доклади от международна конференция, НВУ "Васил Левски", 14-15 юни 2018, Том 9, стр. 167-176.

Цитира се в:

4 Ivaylo Blagoev, G. Vassileva and V. Monov, "Methodology for content preparation of online courses," 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-4, doi: 10.1109/ICAI50593.2020.9311364.

5 Dineva, K., Atanasova, T.: Security in IoT Systems. Proceedings 19th International Multidisciplinary Scientific Geoconference SGEM 2019, 19, 2.1, International Multidisciplinary Scientific Geoconference SGEM, 2019, ISBN:978-

619-7408-79-9, ISSN:1314-2704, DOI:10.5593/sgem2019/2.1, 576-577. SJR (Scopus):0.232 Q4

Участие в проекти

1 Национална научна програма „Информационни и комуникационни технологии за единен цифров пазар в науката, образованието и сигурността“ (ИКТ в НОС) - 2018-2021.

2 Проект Зора по Заповед Но 147/14.06.2019 "Цифров и кибер устойчив ИИКТ"

Награди

1. Награда на ИИКТ-БАН за отлични научни постижения през 2019 г. в категория „Докторанти“.

Декларация за оригиналност на резултатите

Декларирам, че дисертацията съдържа оригинални резултати, получени, при проведени от мен, научни изследвания с подкрепата и съдействието на научния ми ръководител. Резултатите, които са получени, описани и/или публикувани от други учени, са коректно и подробно цитирани в библиографията. Настоящият дисертационен труд не е прилаган за придобиване на научна степен в друго висше училище, университет или научен институт.

Подпис: 

Библиография

1 Adhikari R., Agrawal R. K., An Introductory Study on Time Series Modeling and Forecasting, LAP LAMBERT Academic Publishing, January 29, ISBN 978-3659335082 (2013)

2 Alexandrov, A., Ad-hoc Kalman filter based fusion algorithm for real-time wireless sensor data integration. Flexible Query Answering Systems 2015. AISC, vol. 400, pp. 151–159. Springer, Cham https://doi.org/10.1007/978-3-319-26154-6_12. ISBN 978-3-319-26153-9 (2016)

3 Atanasova T., N. Bakanova, I. Blagoev, Analysis Of Data From Ois To Discover And Model Process-Oriented Information, Сборник доклади от межд. конференция, НВУ "Васил Левски", Том 9, стр. 106-111 (2018)

4 Atanasova, T., Barova, M., Balabanov, T., Using neural models to analyze time series in large volumes of data, Int. Conf. NVU “Vasil Levski”, 193--198, ISSN:1314-1937, (2016)

5 Atanasova, T., Barova, M.: Exploratory analysis of Time Series for hypothesizes feature values. In: International Scientific Conference UniTech 2017, vol. II, pp. 399-403, University publishing house V. Aprilov, Gabrovo (2017)

6 Azoff E. M., Neural Network Time Series Forecasting of Financial Markets, John Wiley & Sons, Inc., 605 Third Ave. New York, NY United States, ISBN:978-0-471-94356-3 (1994)

7 Badrignans, B., Danger, J.-L., Fischer, V., Gogniat, G., Torres, L. (Eds.): Security Trends for FPGAS - From Secured to Secure Reconfigurable Systems. Springer Netherlands (2011)

8 Balabanov T.D., Blagoev I.I., Dineva K.I. Self Rising Tri Layers MLP for Time Series Forecasting. In: Vishnevskiy V., Kozyrev D. (eds) Distributed Computer and Communication Networks. DCCN 2018. Communications in Computer and Information Science, vol 919. Springer, Cham. https://doi.org/10.1007/978-3-319-99447-5_50 (2018)

9 Balabanov, T., Atanasova, T., Blagoev, I., Activation Function Permutation for Multilayer Perceptron Training, International Conference on Big Data, Knowledge and Control Systems Engineering BdKCSE'2018, Sofia, Bulgaria, ISSN 2367-6450, pp. 9-14 (2018)

10 Balabanov, T., Zankinski, I., Dobrinkova, N.: Time series prediction by artificial neural networks and differential evolution in distributed environment. In: Lirkov, I., Margenov, S., Wa´sniewski, J. (eds.) LSSC 2011. LNCS, vol. 7116, pp. 198–205. Springer, Heidelberg https://doi.org/10.1007/978-3-642-29843-1_22. ISBN 978-3-642-29842-4 (2012)

11 Balabanov, T.: Long short term memory in MPL pair. In: Proceedings of the International Scientific Conference UniTech17, Gabrovo, Bulgaria, vol. 2, pp. 375–379 ISSN 1313-230X (2017)

12 Bartneck C. et al.,An Introduction to Ethics in Robotics and AI,SpringerBriefs in Ethics,https://doi.org/10.1007/978-3-030-51110-4_25 (2021)

13 Baumer B. S., Kaplan D. T., Nicholas J., Modern Data Science with R, Horton Chapman & Hall/CRC, Boca Raton (2017)

14 Bernal A., S. Fok, R. Pidaparthi, Financial Market Time Series Prediction with Recurrent Neural Networks (2012)
15 Blagoev I., Dokev N.: A Method for Investigating the Alterations in the Price Trends of the Currency Markets and Forecasting of Probable Future Alterations, *Problems of Engineering Cybernetics and Robotics*, vol.65, pp.39-48 (2012)
16 Blagoev I., Neglected Cybersecurity Risks in the Public Internet Hosting Service Providers. *Information&Security International Journal - ISIJ*, 47, no. 1, pp. 62-76 https://doi.org/10.11610/isij (2020)
17 Blagoev I.: Method for Evaluating the Vulnerability of Random Number Generators for Cryptographic Protection in Information Systems. In: Dimov I., Fidanova S. (eds) Advances in High Performance Computing. HPC 2019. Studies in Computational Intelligence, vol 902. Springer, Cham. https://doi.org/10.1007/978-3-030-55347-0_33. (2021)
18 Blagoev, I., Using R Programming Language for Processing of Large Data Sets, Proc. Int. Conf. Big Data, Knowledge and Control Systems Engineering – BdKCSE'2018, 21-22 November 2018, Sofia, Bulgaria ISSN 2367-6450, pp. 91-98.
19 Borcard, D., Gillet, F., Legendre, P. Numerical Ecology with R, Springer, pp. 9 – 30 (2011)
20 Box G., Jenkins G., and G. Reinsel. Time series analysis: forecasting and control, volume 734. Wiley, (2011)
21 Brockwell P.J. and Davis R.A. Time Series: Theory and methods. Springer. (1991)
22 Brockwell P.J. and Davis R.A., Introduction to Time Series and Forecasting. Springer. (2002)
23 Brown R. G., Brown's General Tools Page: https://phy.duke.edu/
24 Brown R. G.: Dieharder: A Random Number Test Suite, https://webhome.phy.duke.edu/~rgb/General/dieharder.php
25 Bulkowski T., Encyclopedia of Chart Patterns, Wiley (2000)
26 Camara C., Martín H., Peris-Lopez P., Aldalaien M., Design and Analysis of a True Random Number Generator Based on GSR Signals for Body Sensor Networks, *Sensors* 19, 2033; doi:10.3390/s19092033 (2019)
27 Carr J., Simple random number generation, *Computers & Geosciences,* 29(10):1269-1275 (2003)
28 Casti J., I know what you'll do next summer. *New Scientist*, p. 29. (2002).
29 Chen C.M., Jyan H.W., Chien S.C., Jen H.H., Hsu C.Y., Lee P.C., Lee C.F., Yang Y.T., Chen M.Y., Chen L.S., Chen H.H., Chan C.C., Containing COVID-19 Among 627,386 Persons in Contact With the Diamond Princess Cruise Ship Passengers Who Disembarked in Taiwan: Big Data Analytics, *J Med Internet Res*;22(5):e19540, https://www.jmir.org/2020/5/e19540, DOI: 10.2196/19540 (2020)
30 Ciampi F., Marzi G., Demi S., Faraoni M., The big data-business strategy interconnection: a grand challenge for knowledge management. A review and future perspectives, *Journal of Knowledge Management,* Vol. 24, Issue 5 (2020).
31 Cloostermans B., Quasi-linear GCD computation and factoring RSA moduli, Eindhoven University of Technology, Department of Mathematics and Computer Science, Bachelor Mathematics (2012)
32 Cloud Computing – CLOUD 2019, Da Silva D., Dilma, Wang Q., Zhang L.-J. (Eds), 12th International Conference, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, (2019)

33 Creswell W., Research design Qualitative, Quantitative and Mixed Methods approaches, SAGE Publications, (2009)
34 Dalkey, N.C., Delphi. P-3704, RAND Corporation. Santa Monica. CA (1967)
35 Desai1V.V., Patil R. T., Deshmukh V.B.,Rao D. H., Pseudo random number generator using time delay neural network, *World Journal of Science and Technology*, 2(10):165-169 (2012)
36 DiCarlo D., Random Number Generation: Types and Techniques, Liberty University, (2012)
37 Dichtl, M. How to predict the output of a hardware random number generator. CHES 2003, 2779, 181-188. (2003)
38 Diggle P., Time Series. Clarendon Press (1990)
39 Dineva, K.; Atanasova, T., Security in IoT Systems. 19th International Multidisciplinary Scientific GeoConference SGEM 2019, book 2.1, 569-578 (2019)
40 Douc R., Moulines E., Stoffer D. S., Nonlinear time series. Theory, methods and applications with R examples, Chapman and Hall/CRC, New York, ISBN: 978-0429112638 (2014)
41 Ergün S., Security analysis of a chaos-based random number generator for applications in cryptography, 15th International Symposium on Communications and Information Technologies (ISCIT), pp. 319-322, doi: 10.1109/ISCIT.2015.7458371 (2015)
42 Falat L., Stanikova Z., Durisova M., Holkova B., Potkanova T., Application of Neural Network Models in Modelling Economic Time Series with Non-constant Volatility, Procedia Economics and Finance, Volume 34, Pages 600-607 (2015)
43 Falat L., Marcek D., Durisova M., Intelligent Soft Computing on Forex: Exchange Rates Forecasting with Hybrid Radial Basis Neural Network, Hindawi Publishing Corporation, *The Scientific World Journal*, Article ID 3460293 (2016)
44 Fan F., Ge Wang, Learning from Pseudo-Randomness with an Artificial Neural Network – Does God Play Pseudo-Dice? https://arxiv.org/ftp/arxiv/papers/1801/1801.01117.pdf (2018)
45 Four main languages for Analytics, Data Mining, Data Science, http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html (2014)
46 George K., Mutalik P., A Multiple Model Approach to Time-Series Prediction Using an Online Sequential Learning Algorithm, in *IEEE Transactions on Systems, Man, and Cybernetics*: Systems, vol. 49, no. 5, pp. 976-990, May, doi: 10.1109/TSMC.2017.2712184 (2019)
47 Golyandina N., Zhigljavsky A., Singular Spectrum Analysis for Time Series, Springer, Berlin, Heidelberg, 978-3-662-62435-7 (2020)
48 Grover, V.; Lindberg, A.; Benbasat, I.; and Lyytinen, K. The Perils and Promises of Big Data Research in Information Systems, *Journal of the Association for Information Systems*: Vol. 21: Iss. 2, Article 9. DOI: 10.17705/1jais. https://aisel.aisnet.org/jais/vol21/iss2/9 (2020)
49 Hancock, J.T., Khoshgoftaar T.M., CatBoost for big data: an interdisciplinary review. *J Big Data* 7, 94 https://doi.org/10.1186/s40537-020-00369-8 (2020).
50 Hart J. D., Roy R. and Murphy T. E., Optical random number generation - harvesting entropy from noise and chaos, 51st Annual Conference on Information Sciences and Systems (CISS), doi: 10.1109/CISS.2017.7926165 (2017)

51 Hasan, M.M., Popp, J. & Oláh, J. Current landscape and influence of big data on finance. *J Big Data* 7, 21 https://doi.org/10.1186/s40537-020-00291-z (2020)
52 Heaton, J.: Encog Machine Learning Framework. Heaton Research, Inc. http:// www.heatonresearch.com/encog
53 Heninger N., Durumeric Z., Wustrow E., Halderman J. A., Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices, 21st Security Symposium Security'12, ISBN 978-931971-95-9, Bellevue, WA, pp. 205—220 (2012)
54 Hill T., M. O'Connor, W. Remus, Neural Network Models for Time Series Forecasts, *Management Science*, Vol. 42, No. 7, pp. 1082-1092 (1996)
55 Hornik K., M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, vol. 2, no. 5, pp. 359–366, (1989).
56 Iffat A. Gheyas, Leslie S., Smith A., Neural Network Approach to Time Series Forecasting, Proceedings of the World Congress on Engineering, London, U.K., Vol. II WCE 2009, (2009)
57 Jang-Jaccard J., Nepal S., A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, 80(5), 973-993 (2014)
58 Jin, A., Ling, D., Goh A., Biohashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition,* 37, 2245- 2255. (2004)
59 Ketipov R., Kolev K., Sevova J., Blagoev I., Petrov P., Kostadinov G., Zankinski I., Trend and Seasonality Removal with Differential Evolution, *Information Technologies and Control,* Print ISSN 1312 – 2622, Online ISSN: 2367-5357, No. 4, pp.17-22 (2018)
60 Kihoro J.M., Otieno R.O., Wafula C., Seasonal Time Series Forecasting: A Comparative Study of ARIMA and ANN Models, *African Journal of Science and Technology (AJST)* Science and Engineering Series, Vol. 5, No. 2, pages: 41-49 (2004)
61 Kirkpatrick Ch. D., Dahlquist J. R., Technical Analysis: The Complete Resource for Financial Market Technicians, ISBN-13: 978-0131531130 (2006)
62 Koeune F. Pseudo-random number generator. In: van Tilborg H.C.A. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA . https://doi.org/10.1007/0-387-23483-7_330 (2005)
63 Kostadinov G., Atanasova T., Security Policies for Wireless and Network Infrastructure. *Problems of Engineering Cybernetics and Robotics*, vol. 71, 14-19, Bulgarian Academy of Sciences (2019)
64 L'Ecuyer P., Random Number Generation, In book: Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice (2007)
65 Labrinidis A., H. V. Jagadish, Challenges and opportunities with big data, Proceedings of the VLDB Endowment, https://doi.org/10.14778/2367502.2367572 (2012)
66 Lavasani, A., Eghlidos, T. Practical next bit test for evaluating pseudorandom sequences. *Scientia Iranica*, 16(1), 19-33 (2009)
67 Li C., Zhang J., Sang L., Gong L., Wang L., Wang A., Wang Y., Deep Learning-Based Security Verification for a Random Number Generator Using White Chaos, *Entropy,* 22, 1134; doi:10.3390/e22101134 (2020)
68 Long C. (Ед.) Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, Inc. (2015)

69 Maciel L.S., Ballini, R. Design A Neural Network For Time Series Financial Forecasting: Accuracy And Robustness Analisys, http://www.cse.unr.edu/□harryt/CS773C/Project/895-1697-1-PB.pdf (2009)
70 Mahmud M. S., J. Z. Huang, S. Salloum, T. Z. Emara and K. Sadatdiynov, A survey of data partitioning and sampling methods to support big data analysis, in *Big Data Mining and Analytics,* vol. 3, no. 2, pp. 85-101, June 2020, doi: 10.26599/BDMA.2019.9020015 (2020)
71 Martínez-Acosta L., Medrano-Barboza J.-P., López-Ramos Á., López J., López-Lambraño Á., SARIMA Approach to Generating Synthetic Monthly Rainfall in the Sinú River Watershed in Colombia, *Atmosphere,* 11, 602; doi:10.3390/atmos11060602 (2020)
72 Masood F., Faridi A. R., An Overview of Distributed Ledger Technology and its Applications, *International Journal of Computer Sciences and Engineering*, 6(10):422-427, doi:10.26438/ijcse/v6i10.422427 (2018)
73 McKinsey & Company: Perspectives on transforming cybersecurity. Digital McKinsey and Global Risk Practice, March (2019)
74 Mikalef P., Krogstie J., Examining the interplay between big data analytics and contextual factors in driving process innovation capabilities, *European Journal of Information Systems,* Volume 29, - Issue 3: Business Process Management and Digital Innovation https://doi.org/10.1080/0960085X.2020.1740618 (2020)
75 Montgomery D. C., Jennings C. L., Kulahci M., Introduction to Time Series Analysis and Forecasting, Wiley series in probability and statistics, John Wiley & Sons, ISBN 978-0-4 71-65397-4 (2008)
76 Oancea B., Şt. Cr. Ciucu, Time Series Forecasting Using Neural Networks, Challenges of the Knowledge Society, *IT in Social Sciences*, pp.1402-1408 (2013)
77 Oomens W., Maes J., Hasselman F., Egger J., A Time Series Approach to Random Number Generation: Using Recurrence Quantification Analysis to Capture Executive Behavior, *Methods*, Vol. 9, Article 319 (2015)
78 Patil S., Big Data Analytics Using R, *International Research Journal of Engineering and Technology (IRJET)*, Volume: 03, Issue: 07, pp. 78-81 (2016)
79 Person J.L., Candlestick and Pivot Point Trading Triggers, John Wiley & Sons, ISBN 978-0-471-98022-3 (2007)
80 Plummer T., Forecasting Financial Markets: Technical Analysis and the Dynamics of Price (1991)
81 Plummer T., Forecasting Financial Markets: The Psychology of Successful Investing, January (2010)
82 Prechter Robert R., Elliott Wave Principle: Key to Market Behavior, February, (2005)
83 Prechter Robert R., Socionomics: The Science of History and Social Prediction, ISBN-100932750575, April 10, (2003)
84 Prechter Robert R., The Socionomic Theory of Finance, December 28, (2016)
85 Pseudo-Random Number Generators, https://crypto.stanford.edu/pbc/notes/crypto/prng.html
86 Pseudo-Random Number Generators, https://crypto.stanford.edu/pbc/notes/crypto/prng.html
87 Rajaraman V., Big Data Analytics, RESONANCE Proceedings of the Indian Academy of Sciences, August, 695-716. (2016)
88 Random Number Service, https://www.random.org

89 Razzak, M.I., Imran, M. & Xu, G. Big data analytics for preventive medicine. *Neural Comput & Applic* 32, 4417–4451 https://doi.org/10.1007/s00521-019-04095-y (2020)
90 Reinert G., Time Series, Department of Statistics, University of Oxford, http://www.stats.ox.ac.uk/~reinert/time/notesht10short.pdf (2010)
91 Riahi Y., Riahi S.,Big Data and Big Data Analytics: Concepts, Types and Technologies, *International Journal of Research and Engineering* ISSN: 2348-7860 2348-7852 Vol. 5 No. 9 September-October PP. 524-528 (2018)
92 Righetti F., Vallat C., Anastasi G., IoT Applications in Smart Cities: A Perspective Into Social and Ethical Issues, 2018 IEEE International Conference on Smart Computing, IEEE International Conference on Smart Computing (SMARTCOMP), 387-392 DOI: 10.1109/SMARTCOMP.2018.00034 (2018)
93 Ryabko B., Astola J., Malyutov M., Compression-Based Methods of Statistical Analysis and Prediction of Time Series, Springer International Publishing Switzerland, eBook ISBN 978-3-319-32253-7 (2016)
94 Scott G., Carr M., Cremonie M., Technical Analysis: Modern Perspectives, e CFA Institute Research Foundation (2016)
95 Shabri A., Comparison of Time Series Forecasting Methods Using Neural Network and Box-Jenkins Model, *Jurnal Matematika University Teknologi Malaysia*, Jilid 17 bil. 1, hlm. 25-32. (2001)
96 Shamsuddin C.M., Sallehuddin P., Yusof H.M., Artificial Neural Network Time Series Modeling for Revenue Forecasting, *Chiang Mai J. Sci.*; 35(1): (2008)
97 Shumway R.H. and Stoffer D.S., Time Series Analysis and Its Applications. With R Examples. 2nd edition. Springer. (2006)
98 Silipo R. and Phil Winters, „Big Data, Smart Energy, and Predictive Analytics - Time Series Prediction of Smart Energy Data", KNIME.com AG, (2013)
99 Singh S., Maakar S. K. and Kumar S., A Performance Analysis of DES and RSA Cryptography, *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, Volume 2, Issue 3, ISSN 2278-6856 (2013)
100 Slavakis K., G. B. Giannakis, and G. Mateos, Modeling and Optimization for Big Data Analytics, *IEEE Signal Processing Magazine*, September, pp.18-31 (2014)
101 Smith R.L., Time Series. At http://www.stat.unc.edu/faculty/rs/s133/tsnotes.pdf (2001)
102 Staykov B., Andonov F., Practical decision making. *Information Technologies and Control*, Vol 4. ISSN: 1312-2622; Online ISSN: 2367-5357, DOI: 10.1515/itc-2018-0019, pp. 29-38. (2018)
103 Tashev T.D., Hristov H.R. Modeling and Synthesis of Information Interactions. *Problems of Engineering Cybernetics and Robotics*, 52, pp. 75-80 (2001)
104 Tashev, T., Hristov, H.: Modeling of synthesis of information processes with generalized nets. In: Drinov, M. (ed.) *Cybernetics and Information Technologies*, vol. 2, pp. 92–104. Academic Publishing House, Sofia (2003)
105 Tashev, T., Monov, V.: Large-Scale Simulation of Uniform Load Traffic for Modeling of Throughput on a Crossbar Switch Node. In: 8-th Int. Conf. "Large-Scale Scientific Computations" 6-10 June 2011, Sozopol, Bulgaria. LNCS, vol. 7116, pp.630-637. Springer (2012)
106 Technical Analysis https://www.investopedia.com/

107Teh J. S., Alawida M., Sii Y. Ch., Implementation and practical problems of chaos-based cryptography revisited, *Journal of Information Security and Applications*, Volume 50, 102421, ISSN 2214-2126 (2020)
108The R Journal, ISSN: 2073-4859, https://journal.r-project.org/ (2017)
109Tomov, P., Monov, V., Artificial Neural Networks and Differential Evolution Used for Time Series Forecasting in Distributed Environment, Proceedings of International conference „Automatics and Informatics“, ISSN 1313-1850, pp.129-132, Sofia, Bulgaria, Proceedings ISSN 1313-1850, CD ISSN 1313-1869 (2016)
110Toomey D., R for Data Science, Packt Publishing, Birmingham, UK (2014)
111Trappe, L., Washington, L. Introduction to cryptography with coding theory (2nd ed). Upper Saddle River, NJ: Pearson (2006)
112Venables W.N., D.M. Smith and the R Development Core Team, An Introduction to R, Available at https://cran.r-project.org/doc/manuals/R-intro.html#Top (2016)
113Wafi A.S., Hassan H., Mabrouk A., Fundamental Analysis Models in Financial Markets – Review Study, *Procedia Economics and Finance* 30, 939 – 947. Elsevier (2015)
114Wang X., L. T. Yang, Y. Wang, L. Ren and M. J. Deen, ADTT: A Highly Efficient Distributed Tensor-Train Decomposition Method for IIoT Big Data, in *IEEE Transactions on Industrial Informatics,* vol. 17, no. 3, pp. 1573-1582, March 2021, doi: 10.1109/TII.2020.2967768 (2021)
115Wang, W., Y. Wang, Analytics in the era of big data: The digital transformations and value creation in industrial marketing, *Industrial Marketing Management,* Vol. 86, pp. 12-15, ISSN 0019-8501, https://doi.org/10.1016/j.indmarman.2020.01.005 (2020)
116Wickham H., R for Data Science, Garrett Grolemund O'Reilly Publ., Canada (2016)
117 Wollstadt P., Martínez-Zarzuela M., Vicente R., Díaz-Pernas F. J., Wibral M., Efficient Transfer Entropy Analysis of Non-Stationary Neural Time Series, *PLOS ONE*, https://doi.org/10.1371/journal.pone.0102833 (2014)
118Wu D., Wang X., Su J., Tang B., Wu Sh., A Labeling Method for Financial Time Series Prediction Based on Trends, *Entropy*, 22, 1162 (2020)
119Wu X., X. Zhu, G. Wu and W. Ding, Data mining with big data, in *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 1, pp. 97-107, Jan. 2014, doi: 10.1109/TKDE.2013.109. (2014)
120Zankinski, I., Tomov, P., Balabanov, T., Alternative Activation Function Derivative in Artificial Neural Networks, Proceedings of XXV International Symposium Management of Energy, Industrial and Environmental Systems, ISSN 1313-2237, Bankya, Bulgaria, pp. 79-81 (2017)
121Zhang G.P. Neural Networks for Time-Series Forecasting. In: Rozenberg G., Bäck T., Kok J.N. (eds) Handbook of Natural Computing. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-92910-9_14 (2012)
122Zhao P., R with Parallel Computing from User Perspectives, https://www.r-bloggers.com/r-with-parallel-computing-from-user-perspectives/ (2016)

Приложения

Приложение 1. Резултати от RNG теста на Dieharder:

Резултати от анализ на данни събрани от генератор на случайни числа, чрез специализираният софтуер с отворен код Dieharder на Робърт Г. Браун от Физическия факултет на университета Дюк (Brown, 2021):

```
rng_name    |           filename             |rands/second|
 file_input_raw|                  sh-random.bin|  6.02e+07  |
#=============================================================================
=====#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================
=====#
# The file file_input_raw was rewound 52 times
   diehard_birthdays|   0|       100|     100|0.38951753|  PASSED
# The file file_input_raw was rewound 434 times
      diehard_operm5|   0|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 922 times
  diehard_rank_32x32|   0|     40000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 1151 times
    diehard_rank_6x8|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 1251 times
   diehard_bitstream|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 2051 times
        diehard_opso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 2584 times
        diehard_oqso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 2834 times
         diehard_dna|   0|   2097152|     100|0.00020081|   WEAK
# The file file_input_raw was rewound 2859 times
diehard_count_1s_str|   0|    256000|     100|0.01953386|  PASSED
```

```
# The file file_input_raw was rewound 3347 times
diehard_count_1s_byt|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 3356 times
 diehard_parking_lot|   0|     12000|     100|0.33358085|  PASSED
# The file file_input_raw was rewound 3362 times
    diehard_2dsphere|   2|      8000|     100|0.03334174|  PASSED
# The file file_input_raw was rewound 3367 times
    diehard_3dsphere|   3|      4000|     100|0.92330162|  PASSED
# The file file_input_raw was rewound 4246 times
     diehard_squeeze|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 4246 times
       diehard_sums|   0|       100|     100|0.14880589|  PASSED
# The file file_input_raw was rewound 4284 times
       diehard_runs|   0|    100000|     100|0.00000004|  FAILED
       diehard_runs|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 4797 times
      diehard_craps|   0|    200000|     100|0.00000000|  FAILED
      diehard_craps|   0|    200000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 12427 times
 marsaglia_tsang_gcd|   0|  10000000|     100|0.00000000|  FAILED
 marsaglia_tsang_gcd|   0|  10000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 12465 times
       sts_monobit|   1|    100000|     100|0.00000222|   WEAK
# The file file_input_raw was rewound 12503 times
         sts_runs|   2|    100000|     100|0.03544040|  PASSED
# The file file_input_raw was rewound 12541 times
        sts_serial|   1|    100000|     100|0.00000278|   WEAK
        sts_serial|   2|    100000|     100|0.00000062|  FAILED
        sts_serial|   3|    100000|     100|0.00198630|   WEAK
        sts_serial|   3|    100000|     100|0.24256527|  PASSED
        sts_serial|   4|    100000|     100|0.00001279|   WEAK
        sts_serial|   4|    100000|     100|0.00000000|  FAILED
```

```
        sts_serial|   5|    100000|     100|0.14830043|  PASSED
        sts_serial|   5|    100000|     100|0.00006248|   WEAK
        sts_serial|   6|    100000|     100|0.01594394|  PASSED
        sts_serial|   6|    100000|     100|0.58120558|  PASSED
        sts_serial|   7|    100000|     100|0.00001243|   WEAK
        sts_serial|   7|    100000|     100|0.00650289|  PASSED
        sts_serial|   8|    100000|     100|0.00000000|  FAILED
        sts_serial|   8|    100000|     100|0.00000000|  FAILED
        sts_serial|   9|    100000|     100|0.00000000|  FAILED
        sts_serial|   9|    100000|     100|0.14200950|  PASSED
        sts_serial|  10|    100000|     100|0.00000000|  FAILED
        sts_serial|  10|    100000|     100|0.00003391|   WEAK
        sts_serial|  11|    100000|     100|0.29281609|  PASSED
        sts_serial|  11|    100000|     100|0.00000000|  FAILED
        sts_serial|  12|    100000|     100|0.10890305|  PASSED
        sts_serial|  12|    100000|     100|0.04145417|  PASSED
        sts_serial|  13|    100000|     100|0.00000000|  FAILED
        sts_serial|  13|    100000|     100|0.00000000|  FAILED
        sts_serial|  14|    100000|     100|0.00000037|  FAILED
        sts_serial|  14|    100000|     100|0.51404682|  PASSED
        sts_serial|  15|    100000|     100|0.32460847|  PASSED
        sts_serial|  15|    100000|     100|0.00000000|  FAILED
        sts_serial|  16|    100000|     100|0.00651735|  PASSED
        sts_serial|  16|    100000|     100|0.00115580|   WEAK
# The file file_input_raw was rewound 12618 times
       rgb_bitdist|   1|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 12770 times
       rgb_bitdist|   2|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 12999 times
       rgb_bitdist|   3|    100000|     100|0.03240048|  PASSED
# The file file_input_raw was rewound 13304 times
       rgb_bitdist|   4|    100000|     100|0.00000000|  FAILED
```

```
# The file file_input_raw was rewound 13686 times
      rgb_bitdist|   5|    100000|     100|0.88959066|  PASSED
# The file file_input_raw was rewound 14143 times
      rgb_bitdist|   6|    100000|     100|0.00000006|  FAILED
# The file file_input_raw was rewound 14677 times
      rgb_bitdist|   7|    100000|     100|0.07126523|  PASSED
# The file file_input_raw was rewound 15288 times
      rgb_bitdist|   8|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 15974 times
      rgb_bitdist|   9|    100000|     100|0.32917367|  PASSED
# The file file_input_raw was rewound 16737 times
      rgb_bitdist|  10|    100000|     100|0.00050227|   WEAK
# The file file_input_raw was rewound 17577 times
      rgb_bitdist|  11|    100000|     100|0.15629093|  PASSED
# The file file_input_raw was rewound 18492 times
      rgb_bitdist|  12|    100000|     100|0.00001785|   WEAK
# The file file_input_raw was rewound 18568 times
rgb_minimum_distance|   2|     10000|    1000|0.00000012|  FAILED
# The file file_input_raw was rewound 18683 times
rgb_minimum_distance|   3|     10000|    1000|0.00000022|  FAILED
# The file file_input_raw was rewound 18836 times
rgb_minimum_distance|   4|     10000|    1000|0.00000000|  FAILED
# The file file_input_raw was rewound 19026 times
rgb_minimum_distance|   5|     10000|    1000|0.00206076|   WEAK
# The file file_input_raw was rewound 19103 times
   rgb_permutations|   2|    100000|     100|0.00012861|   WEAK
# The file file_input_raw was rewound 19217 times
   rgb_permutations|   3|    100000|     100|0.00000003|  FAILED
# The file file_input_raw was rewound 19370 times
   rgb_permutations|   4|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 19560 times
   rgb_permutations|   5|    100000|     100|0.00000000|  FAILED
```

```
# The file file_input_raw was rewound 19942 times
    rgb_lagged_sum|   0|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 20705 times
    rgb_lagged_sum|   1|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21849 times
    rgb_lagged_sum|   2|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 23375 times
    rgb_lagged_sum|   3|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 25282 times
    rgb_lagged_sum|   4|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 27571 times
    rgb_lagged_sum|   5|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 30241 times
    rgb_lagged_sum|   6|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 33293 times
    rgb_lagged_sum|   7|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 36726 times
    rgb_lagged_sum|   8|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 40541 times
    rgb_lagged_sum|   9|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 44737 times
    rgb_lagged_sum|  10|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 49315 times
    rgb_lagged_sum|  11|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 54274 times
    rgb_lagged_sum|  12|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 59615 times
    rgb_lagged_sum|  13|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 65337 times
    rgb_lagged_sum|  14|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 71440 times
    rgb_lagged_sum|  15|   1000000|     100|0.00000000|  FAILED
```

```
# The file file_input_raw was rewound 77925 times
    rgb_lagged_sum|  16|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 84792 times
    rgb_lagged_sum|  17|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 92040 times
    rgb_lagged_sum|  18|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 99669 times
    rgb_lagged_sum|  19|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 107680 times
    rgb_lagged_sum|  20|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 116072 times
    rgb_lagged_sum|  21|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 124846 times
    rgb_lagged_sum|  22|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 134001 times
    rgb_lagged_sum|  23|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 143538 times
    rgb_lagged_sum|  24|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 153456 times
    rgb_lagged_sum|  25|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 163756 times
    rgb_lagged_sum|  26|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 174437 times
    rgb_lagged_sum|  27|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 185500 times
    rgb_lagged_sum|  28|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 196944 times
    rgb_lagged_sum|  29|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 208769 times
    rgb_lagged_sum|  30|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 220976 times
    rgb_lagged_sum|  31|   1000000|     100|0.00000000|  FAILED
```

```
# The file file_input_raw was rewound 233565 times
     rgb_lagged_sum|  32|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 233603 times
    rgb_kstest_test|   0|     10000|    1000|0.02580373|  PASSED
# The file file_input_raw was rewound 234189 times
    dab_bytedistrib|   0|  51200000|       1|0.00000000|  FAILED
# The file file_input_raw was rewound 234238 times
          dab_dct| 256|     50000|       1|0.00000000|  FAILED
Preparing to run test 207.  ntuple = 0
# The file file_input_raw was rewound 234669 times
       dab_filltree|  32|  15000000|       1|0.00000000|  FAILED
       dab_filltree|  32|  15000000|       1|0.00000000|  FAILED
Preparing to run test 208.  ntuple = 0
# The file file_input_raw was rewound 234781 times
      dab_filltree2|   0|   5000000|       1|0.00000000|  FAILED
      dab_filltree2|   1|   5000000|       1|0.00000000|  FAILED
Preparing to run test 209.  ntuple = 0
# The file file_input_raw was rewound 235029 times
       dab_monobit2|  12|  65000000|       1|1.00000000|  FAILED
```